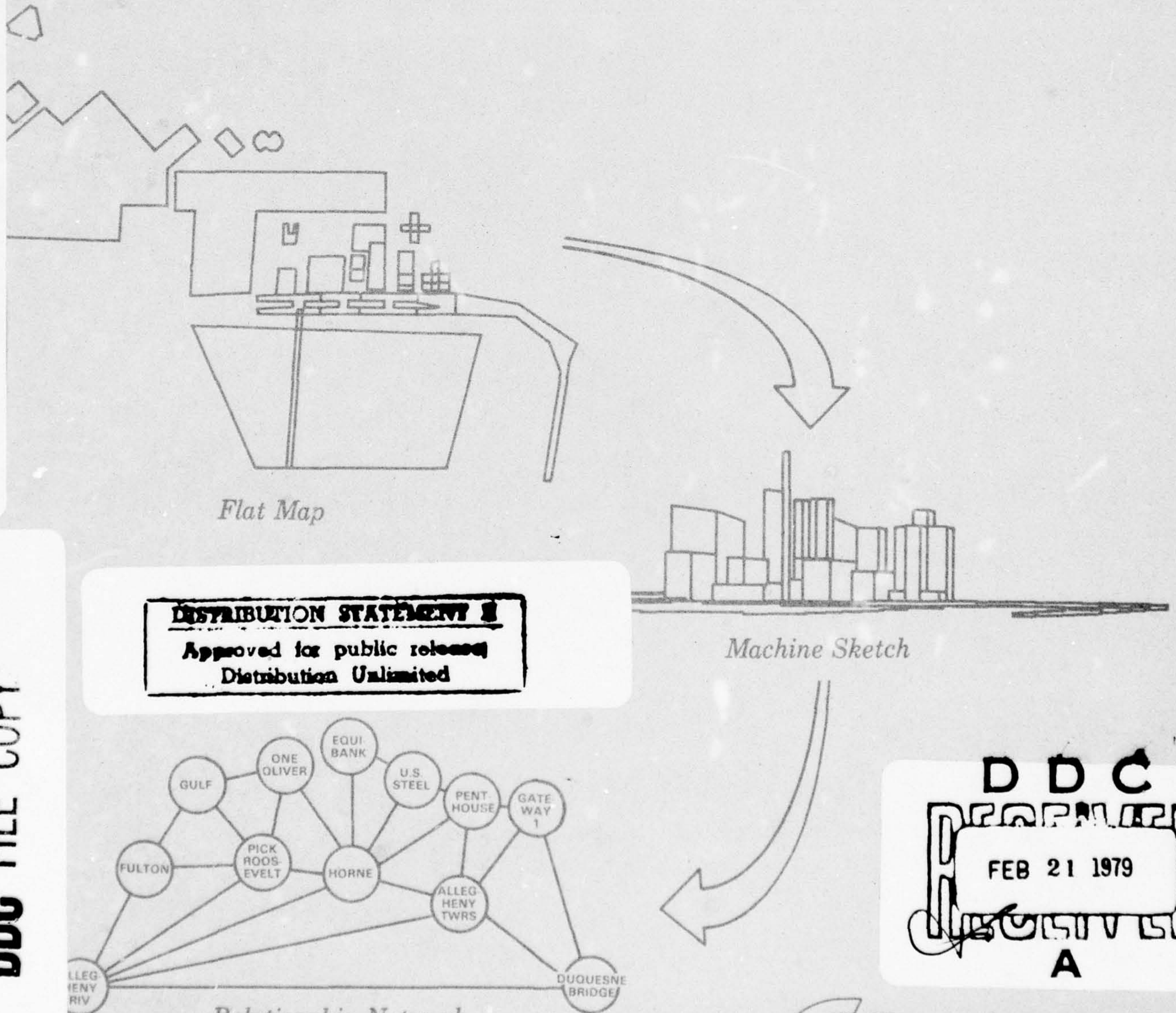
PROCEEDINGS:
IMAGE UNDERSTANDING WORKSHOP

NOVEMBER 1978

Sponsored by:
Information Processing Techniques Office
Defense Advanced Research Projects Agency

Hosted by:
Carnegie-Mellon University
Pittsburgh, Pennsylvania



Flat Map

Machine Sketch

Relationship Network

Science Applications, Inc.

(6)

IMAGE UNDERSTANDING.

Proceedings of a Workshop held at Pittsburgh, Pennsylvania November 14-15, 1978 ;

Sponsored by the
Defense Advanced Research Projects Agency

(9) Semi-Annual Technical rept. May - Nov 78,

(11) Nov 78

(12) 204p.

(14)

Science Applications, Inc.
Report Number SAI-79-814-WA
(10) Lee S. Baumann
Workshop Organizer and
Proceedings Editor

This report was supported by
the Defense Advanced Research
Projects Agency under DARPA
Order No. 3456, contract No. MDA903-78-C-0095,
(15) monitored by the
Defense Supply Service, Washington, D.C.

DARPA Order-3456

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of the Defense Advanced Research Projects Agency or the United States Government.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
SAI-79-814-WA

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Proceedings:
Image Understanding Workshop, November 1978

5. TYPE OF REPORT & PERIOD COVERED
Semiannual Technical
May 1978 - November 1978

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Lee S. Baumann (Ed.)

8. CONTRACT OR GRANT NUMBER(s)
MDA903-78-C-0095

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Science Applications, Inc.
1911 N. Fort Myer Drive, Suite 1200
Arlington, Virginia 22209

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
ARPA Order No. 3456

11. CONTROLLING OFFICE NAME AND ADDRESS
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

12. REPORT DATE
November 1978

13. NUMBER OF PAGES
195

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital Image Processing, Image Understanding, Scene Analysis, Edge Detection, Image Segmentation, CCD Arrays, CCD Processors

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document contains the technical papers and outlines of semi-annual progress reports presented by the research activities in Image Understanding sponsored by the Information Processing Techniques Office; Defense Advanced Research Projects Agency. The papers were presented at a workshop conducted on 14-15 November 1978 in Pittsburgh, Pennsylvania.

TABLE OF CONTENTS

PAGE

SESSION I - PROGRAM REVIEWS BY PRINCIPAL INVESTIGATORS

SESSION II - TECHNIQUES I

TABLE OF CONTENTS (Cont.)

PAGE

SESSION III - TECHNIQUES II

SESSION IV - SYSTEMS

SESSION V - HARDWARE

FORWARD

The Eighth Image Understanding Workshop marked the first to be held under the direction of the new Defense Advanced Research Projects Agency Program Manager, Major Larry E. Druffel of the United States Air Force. Since Major Druffel inherited a mature research program, it was deemed appropriate that he set forth his views concerning the present state and probable future of the program. The following quotation represents Major Druffel's perceptions at this time.

> "As the Image Understanding Program enters its fourth year of a planned five year effort, it is appropriate to examine past progress and future direction. From the inception it was clear that image understanding was a high risk area with an equally high payoff potential. The thrust of the program has been broadly aimed at the development of techniques for a number of possible applications including photointerpretation, cartography, target cueing, navigation, and symbolic bandwidth. In the past three years we have witnessed significant advances which certainly justify the investment. Techniques developed within the program are beginning to find their way into planned military systems. However, because of the broad focus of the program, there is no immediately identifiable emergent system.
>
> A common theme in these workshops has been the need for a concept demonstration. At each workshop, Lt. Colonel Carlstrom reiterated the importance of focus of effort toward a demonstration at the end of five years. It is clear that image problems are difficult and that the likelihood of a demonstration within the next two years is very low. Although giant strides have been made, the road is longer and more difficult than anticipated. The risk does not seem quite so high and the payoff just as great as originally supposed. Continued investment in the research seems warranted. However, if we are to realize eventual payoff, the time for talking about concept demonstration has passed and the time for planning has come.
>
> The prudent approach is to consolidate those techniques which are sufficiently mature for transfer to DoD agencies. The remainder of the program must then be pursued with a narrower focus. This increased focus will take the form of a scenario in which the follow-on research will be constrained. In the next four months, I will be aggressively pursuing the definition of an appropriate focus. Definition of such a scenario will take a great deal of thought and cooperative effort, both from the potential user community and from the research community. The increased focus will not be toward the development of a single system, rather it will be toward the development of the tools needed for inclusion in some future system.
>
> If there is fault with the program, it is the paucity of imagery. The research community has been sadly constrained by the unavailability of appropriate imagery. The researchers are well aware of the need to make their algorithms robust, but verification of this robustness requires application of their techniques over a wide range of imagery. The success of this effort, perhaps even the further existence of this effort, is heavily dependent on cooperation of the using community in providing imagery to support the program. Cooperative participation from both the research community and from the user community is sincerely invited both in recommending a focus and in the development of appropriate imagery."

This document contains the technical reports and program reviews presented by the principal investigators and research personnel at the Eighth Image Understanding Workshop held at Carnegie-Mellon University, Pittsburgh, Pennsylvania, on 14-15 November 1978. In attendance at the workshop, in addition to the University and Industrial research personnel, were representatives from many Army, Navy, Air Force and Government Agency organizations interested in the accomplishments of this research program. The workshop provided the opportunity for a lively exchange of views between the potential user community and those organizations actively pursuing research in Image Understanding.

The workshop was hosted by Dr. D. Raj Reddy, Professor of Computer Science at Carnegie-Mellon University. I wish to express the appreciation of all attendees for the excellent facilities and hospitality which Dr. Reddy so kindly extended to make the workshop a success. The workshop organizer also wishes to thank Mrs. Beverly Howell of the Computer Science Department at CMU for her efforts in making the necessary administrative arrangements for the workshop in Pittsburgh. Also my thanks to Miss Carrie Howell of Science Applications, Inc. for providing typing support for mailings and the collection and arrangement of the conference proceedings.

The cover design was created by Miss Elody Blomberg and Mr. Thomas G. Dickerson of the Art Department of Science Applications, Inc. from material supplied by Dr. Steven Rubin of the Computer Science Department at Carnegie-Mellon University. The sketches are all of the host city of Pittsburgh and are successively used in computer processing by the ARGOS Image Understanding System developed at CMU. Dr. Rubin informs us that the flat map is produced as a first step to impart knowledge to the system. From this knowledge, the ARGOS is able to produce the machine sketch and the relationship network which it will subsequently utilize to identify objects in photographs presented to the system. For a more lucid and more detailed explanation see Dr. Rubin's paper on the ARGOS system.

Lee S. Baumann
Science Applications, Inc.
Workshop Organizer

AUTHOR INDEX

SESSION I

PROGRAM REVIEWS

BY

PRINCIPAL INVESTIGATORS

MIT PROGRESS IN UNDERSTANDING IMAGES

Patrick H. Winston

The Artificial Intelligence Laboratory
Massachusetts Institute of Technology

In this series of image understanding conference proceedings, we have stressed the key issue of representation. In particular, we have described the work of Horn and his collaborators using the reflectance map *and the* albedo image *in working with satellite images, and we have described the work of Marr and his collaborators using the* primal sketch, *the* 2 1/2 D sketch, *and* body-centered, 3-D models *to work toward a comprehensive theory of recognition.*

Here, we begin with a review of the overall program, briefly explaining our approach, stating the objectives, and citing the fundamental tools. Then we summarize the results obtained through an enumeration of representative individual efforts, concentrating on work now in progress.

Marr's View of Vision Theory

Marr has proposed and championed the idea that vision research must follow these steps:

- First, a competence to be understood is precisely described. Often this means understanding the limits of the various modules of the human vision system. Knowing the strength of the various modules in an existing, clearly good system, helps us to know what competence is needed in the modules of the computer-based systems of the future.
- Second, representations are selected or invented that facilitate explicit description of the target processing products.
- Third, the competence and the representations are combined into a well-defined computation problem to be solved.
- Fourth, algorithms are devised that perform the desired computation.
- And fifth, results are validated by computer implementation.

Importantly, Marr believes it is wrong to begin with devotion to some particular type of algorithm with a view toward finding a problem that it will solve.

At the highest level, observation of competences and definition of representations have led Marr to think in terms of the competences and representations suggested in figure 1. As shown, there are three levels of representation. The *primal sketch* makes information about intensity changes explicit, including the length, position, orientation, and contrast of line fragments. The *2 1/2 D* sketch makes information about surface orientation explicit. And the the *3-D model* makes information about object shape explicit.

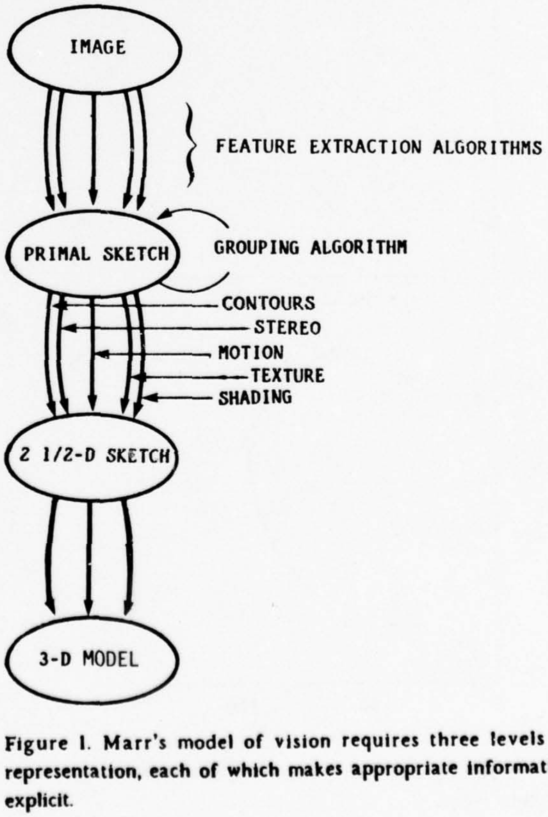


Figure 1. Marr's model of vision requires three levels of representation, each of which makes appropriate information explicit.

Past Results and Current Foci

The early work in Marr's group was largely devoted to specifying the three levels of representation required by the overall theory and to the computation of the primal sketch. (Figure 2 illustrates one step of that computation.) Now emphasis is shifting to the problems involved in going from one representational level to another and in using the primal sketch to deal with texture.

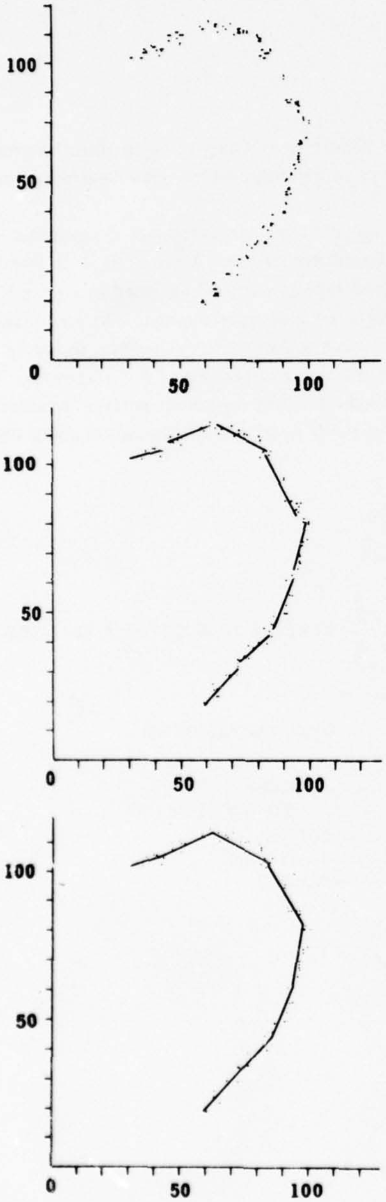


Figure 2. Finding a boundary from the place tokens contained in the primal sketch.

One example of the representation specification work is that of Nishihara on first extending the generalized cylinder representation invented by Binford at Stanford and then on shape recognition.

The key to shape recognition is to produce as consistent a description as possible of shape from the local surface information available in the 2 1/2 D sketch. The description should not, for example, depend on the viewer's vantage point. Marr and Nishihara have stated the problem formally in terms of three criteria, *accessibility*, *scope* and *uniqueness*, and *sensitivity* and *stability*. From this they determined that to be suitable for recognition a shape representation should be (1) based on the arrangement of volumetric features such as centers of mass and axes of elongation or symmetry, (2) that these arrangements should be specified in an object-centered coordinate frame (as opposed to a viewer-centered one like that of the 2 1/2 D sketch), and (3) the description should be modular with each module specifying the relative arrangement of a small number of related features which could stand alone as a shape description. Nishihara's thesis deals with the problem of computing such a description from the 2 1/2 D sketch. The work includes a consideration of a technique based on identifying chains of local ridge points at a given resolution and over a range fixed by the resolution. The results are not complete but early indications are promising and further work is in progress.

An example having to do with getting texture information out of the primal sketch is the work of Stevens on the computation of "flow." His paper on the subject is included in these proceedings. Another is the work of Riley. His demonstration that only simple computations are needed to handle the orientation component of texture analysis is good news for computer vision.

On another front, the work of Ullman on motion is representative of what needs to be done in order to go confidently from the primal sketch to higher level representations. He first showed that it makes sense to match successive views at a low level approximating that of the primal sketch. He then proved a variety of theorems having to do with what, minimally, is required to get the three-dimensional shape of an object out of images of it. (Although Ullman's treatment of the matching problem was of major importance, more remains to be done. Indeed the correspondence problem, as we call it, is occupying a large fraction of the resources of Marr's group at the moment.)

Stereo has also received attention as a way of going upward from the primal sketch. Marr and Poggio, in collaboration, have devised two quite different theories of how to do stereo. The newer one is now undergoing testing and we expect to report on experiments with it by Grimson and Hildreth at the next workshop. Already our initial implementation seems highly succesful in computing disparity from a stereo pair of photographs taken of natural scenes. Currently, we are turning towards issues concerning the "filling in" of depth information where it cannot be recovered directly from the image. These issues interface with more general issues concerning the representation of spatial information.

Still another way of extracting depth information from the primal sketch has to do with using local line orientations and junction angles to postulate surface orientaion. Stevens' work on this problem is now jelling nicely.

Primal Sketch Hardware

Since much of Marr's image understanding work requires the computation of the primal sketch, it is important to be able to compute the primal sketch quickly. This in turn requires an ability to do a great deal of convolution. Thus our new image convolution box, ICON, has become an important factor in pushing research ahead, making possible convolutions of larger images with larger masks in a reasonable amount of time.

ICON combines a pipelined VLSI multiplier with a fast bipolar image cache. Approximately 120 Schottky MSI and LSI IC's are used. The device is connected as a peripheral to the LISP Machine and is driven by microcode. It performs its job on the order of 100 times faster than our PDP-10 for only a few thousand dollars in hardware cost.

The software on the LISP Machine which drives the box makes it possible to handle masks that are larger than the convolver's 1024 point fast internal memory by breaking them up into manageable chunks and adding together their results. Additionally, the software puts resolution under simple program control by allowing users to specify which points in an image the convolver is to be run on.

Based on our experience with ICON, we are beginning to plan the design of another convolution box. This device will have more memory and will be faster.

Horn Concentrates on Understanding Image Formation

Understanding an image implies a need to understand how light reflection depends on various combinations of surface material, surface orientation, and light-source position. Among the products are tools for dealing with the following needs:

- Automated generation of shaded relief maps.
- Generation of low-level, obliquely-viewed images.
- Generation of special maps that bring out particular terrain features.
- Classification of ground cover for crop prediction.
- Matching images to terrain data for satellite navigation.
- Making maps for automatic or semiautomatic change detection.

The roadmap for the theory development is shown in figure 3. As shown, the progression again involves a number of key representations: the reflectance map, the digital terrain map, the synt..etic image, the multiple-sun synthetic image, the albedo image, and the change-detection image. Since understanding reflectance maps is prerequisite to following Horn's work, we now describe what is involved.

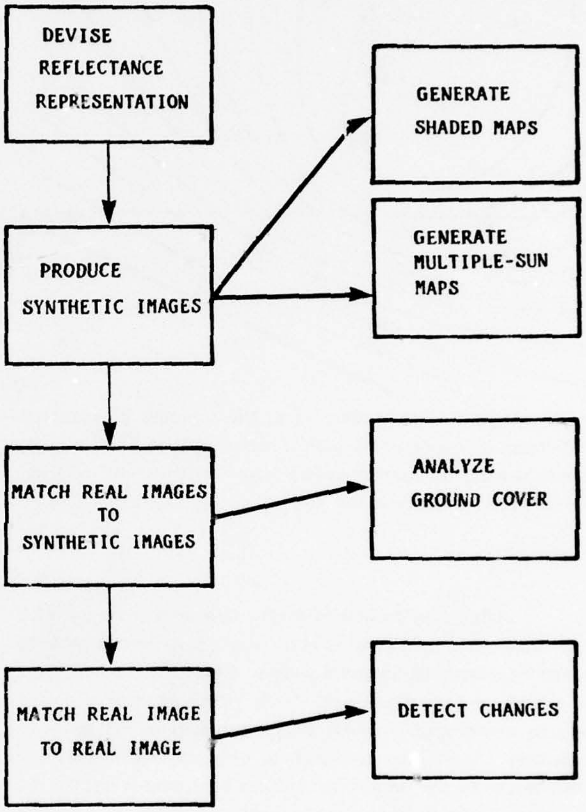


Figure 3. Roadmap for the development of a theory of image formation and exploitation. Some applications of the theory appear to the right.

The purpose of the reflectance map is to make explicit the relationship among observed intensity, surface material, surface orientation, and light-source position. To see how, consider figure 4. All points (p, q) in the space correspond to surface orientations. For a given surface material and light-source position, a surface's orientation determines its reflected light intensity. By drawing lines through points representing orientations that have the same intensity, one gets the isointensity lines shown. This particular map is for illumination from the upper left.

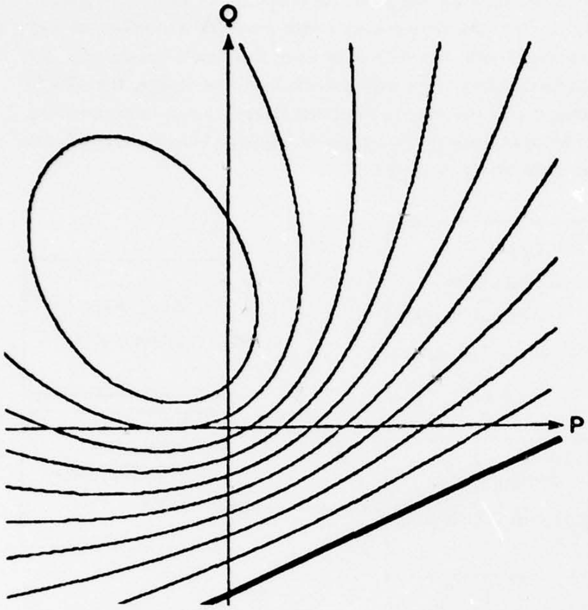


Figure 4. In this reflectance map, the contours of constant reflectance correspond to a normal surface material and a light source striking the viewed surface from the upper left (or from the northwest, thinking in map terms).

Once it is possible to predict intensities from material, orientation, and light-position information, it is then possible to produce synthetic high-altitude images. Figure 5 shows an image of a piece of Switzerland synthetically generated using a digital terrain model and a simple reflectance-map model of light reflection. Appropriate combinations of ground cover and sun position can be used to give the user the best possible feel for the mountains and hills that constitute the terrain.

Interestingly, however, shaded relief maps need not conform to what might actually be observed. Horn has made images that correspond to terrain illuminated by three suns, one blue, one red, and one green. Such images give special insight into terrain properties at a glance. Slopes with exposure to the south, for example, are readily identified because of their red hue from the red, southern sun.

The thrust of Horn's work, however, is to make images that match photographs as closely as possible with a view toward registering real aerial photographs with terrain models. Such matching is a vital first step toward improving the use of satellite images.

After a real aerial photograph is registered with a synthetic one produced from a terrain model, some areas will refuse to match well because the actual ground cover is not the one assumed in generating the synthetic image. Horn defines an *albedo map* to be an image in which each point's intensity is the ratio of the intensity in the real image to the intensity in the synthetic image. In addition to use in classification, it seems likely that albedo maps will be useful in change detection. It would be nice if change could be detected by subtracting one image from another. Unfortunately, the changes in sun position from hour to hour and from day to day make this impossible by swamping changes caused by changes in the ground cover. Instead, Horn proposes to divide earlier and later real image intensities by the intensities predicted by the terrain model to give two registered albedo maps. Then, one albedo map is subtracted from the other, producing change that will correspond to ground-cover differences occuring between the earlier and later recording times.

For human use, the two albedo maps can be printed in different colors and superimposed. The human analyst's eye is instantly drawn to places where changes have taken place because their hue will differ from the surrounding area.

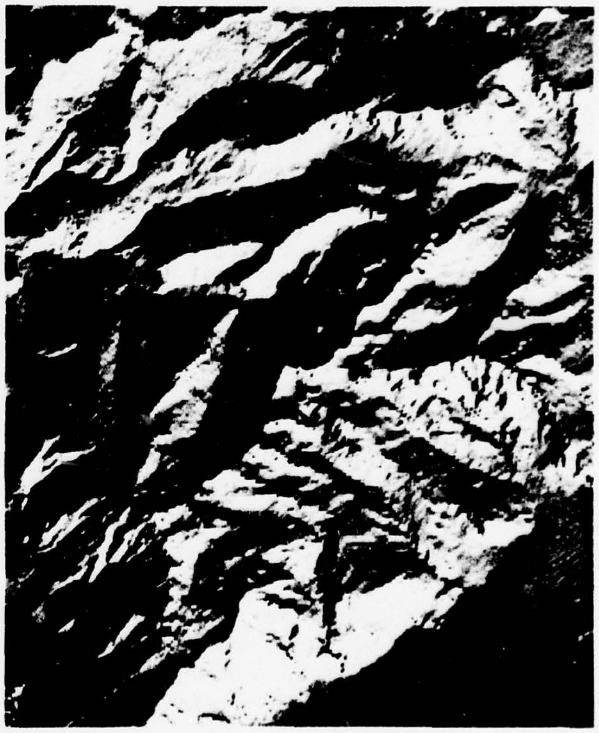


Figure 5. A synthetic image of mountainous terrain.

Making Good Synthetic Images Requires Attention to Many Details

To make really useful synthetic images, we have found it necessary to solve several subproblems of the sort that escape notice when thinking is done in terms of idealized domains. One of these is the problem of introducing cast shadows into the synthetic image. This has been done.

Other problems include those introduced by the characteristic flaws of satellite images, by the need for care in dealing with coordinate transformations, and by the need to know accurately where the sun is. Horn's group has developed straightforward methods for dealing with all three of these problems.

Of the three, perhaps the most interesting has to do with the corrections to satellite images that must be made to account for differences in the transfer functions of the several sensors used. The preceding proceedings included a paper by Horn and Woodham that gives the results of their work on the problem. The paper describes a method that uses statistics obtained from the sensors themselves, together with an assumption that the probability distribution of the scene radiance seen by each image sensor is the same. Using this method, they have sucessfully removed the striping effects seen commonly in satellite photographs.

Representative Recent Results

Horn's group has been working at a furious pace, producing new papers on a number of subjects.

Horn, Woodham, and Silver, for example, describe a method by which surface orientation can be derived using a fixed sensor together with varying lighting. This method is called photometric stereo inasmuch as it is the complement of ordinary stereo with its use of varying sensor position. Conveniently, the correspondence problem disappears, since a fixed sensor position insures that there is no question about how points in one image correspond to points in another.

Strat, working on image generation rather than image analysis, has described how the architecture of data-flow computation can exploit parallelism to generate shaded images of terrain in less than one-tenth of a second.

And Horn and Sjoberg, in a paper included in these proceedings, give a unified approach to the specification of surface reflectance in terms of both incident and reflected beam geometry. In their paper they derived the reflectance map in terms of the so-called bidirectional reflectance-distribution function used b) the National Bureau of Standards.

Other work, now being documented, includes results obtained by Horn and Strat on the fast computation of shape from shading information. Previously the necessary computations seemed to require numerical integrations along certain image contours. The new results show that a cooperative algorithm can do the same computation in a much faster parallel fashion. Bruss is working out the conditions under which the new algorithm coverges.

At the moment, much attention is going into an effort aimed at atmospheric modeling, with a view toward further improvement of the image matching process already demonstrated.

Ref^rences

For an extended discussion of MIT work on Image Understanding, see volume 2 of *Artificial Intelligence: an MIT Perspective*, edited by Patrick H. Winston and Richard H. Brown, MIT Press, Cambridge, Massachusetts, 1979.

Berthold K. P. Horn and Brett L. Bachman, "Using Synthetic Images to Register Real Images with Surface Models," AIM-437, The Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1977. (To appear in CACM, November, 1978.)

Berthold K. P. Horn and Robert J. Woodham, "LANDSAT MSS Coordinate Transformations," AIM-465, The Artificial Intelligence Laboratory, 1978.

Berthold K. P. Horn, "The Position of the Sun," Working Paper 162, The Artificial Intelligence Laboratory, 1978.

Berthold K. P. Horn and Robert J. Woodham, "Destriping Satellite Images," AIM-467, The Artificial Intelligence Laboratory, 1978. (To appear in CGIP.)

Berthold K. P. Horn and Robert W. Sjoberg, "Calculating the Reflectance Map," AIM-495, The Artificial Intelligence Laboratory, 1978. Also included in these proceedings.

Berthold K. P. Horn, Robert J. Woodham, and William M. Silver, "Determining Shape and Reflectance using Multiple Images," AIM-490, The Artificial Intelligence Laboratory, 1978.

David Marr, "Representing Visual Information," AIM-415, The Artificial Intelligence Laboratory, 1977.

David Marr and Keith Nishihara, "Representation and Recognition of the Spatial Organization of Three-dimensional Shapes," *Phil. Trans. Roy. Soc. B. 275*, in publication.

Kent Stevens, "Computation of Locally Parallel Structure," *Biological Cybernetics*, in publication. Also included in these proceedings.

Thomas M. Strat, "Shaded Perspective Images of Terrain," AIM-463, The Artificial Intelligence Laboratory, 1978.

Thomas M. Strat, "Application of Data Flow Computation to the Shaded Image Problem," Working Paper 163, The Artificial Intelligence Laboratory, 1978.

PROGRESS AT THE ROCHESTER IMAGE UNDERSTANDING PROJECT

C. M. Brown
J. A. Feldman
K. R. Sloan, Jr.

Department of Computer Science
University of Rochester
Rochester, New York 14627

1. Model Refinement

1.1. Procedural Description

One important goal of the Rochester Vision Project is to investigate a generalized form of procedural invocation in which an executive procedure chooses worker procedures to perform a job not just on the basis of input/output behavior (as traditional pattern- directed invocation does), but also taking into account cost/benefit estimates and perhaps other information as well. This scheme is motivated by the desire to have the advantages of declarative knowledge about what is doable (the descriptions) along with the advantages of procedural knowledge about how to do it (the workers). The declarative, descriptive component will allow conviences such as the modular addition of procedural knowledge. The main research issue is to decide what exactly needs to be known about worker procedures, and how to express that in a useful and uniform manner. This must also be coordinated with the use of relational constraints [Russell and Brown, 1978]. The most recent and presently contemplated work at Rochester explores aspects of these issues (e.g. Lantz, Ballard, and Brown, 1978).

1.2. Decision Theory

The use of decision theory not only as an abstract model of intelligent perception but as a practical tool to maximize computational benefit/cost is being investigated in the context of procedural invocation. This work continues in the tradition of Bolles, Sproull, and Garvey, and ultimately we hope to extend some of their results to deal with formal problems that more closely approximate the sorts of vision problems encountered in our particular applications. Ballard (see Section 2) uses decision theory techniques to choose the most economical method (assuring adequate accuracy) of locating anatomical structures in large-format images.

2. Applications in Biomedicine

The model-directed finding of ribs in chest radiographs [Ballard, 1978] provides an illustration of the use of the Rochester Vision System, incorporating procedure description, utility measures, and tops-down, model-directed perception. The object here is to cope with large amounts of possibly low-quality data without undue processing time by depending on a declarative model of anatomical structures, described procedural knowledge about how to locate them, and an executive which uses decision theory to control the image- understanding process. A prototype complete analysis system is now being developed.

A novel and uniform method of describing arbitrary functions on the unit sphere (which define "museum-viewable" volumes) is under investigation, with immediate application to anatomical structures [Schudy 1978]. The idea is related to the well- known Fourier descriptions of two- dimensional shape. Volumes are modelled and described as the leading coefficients in certain spherical harmonic expansions of the volume functions. This method also allows least squared error fitting of volumes in coefficient space, which interfaces nicely with routines which locate the three- dimensional boundaries of volumes in image data.

3. Application in Aerial Image Analysis

The three-level organization of image analysis (strategist, executive, worker) and a further exploration of useful procedural description mechanisms are the objects of study in automatic photo- interpretation work [Lantz 1978].

The object is to use the sorts of knowledge- based inferencing used by skilled photointerpreters, along with models inspired by photointerpretation keys for identifying small industries, to do reliable and flexible identification of a few types of small industrial installations. Imagery has been acquired from a Rochester, N.Y. mapping firm and from RADC in Rome, N.Y.

4. Fast Display of Certain Polyhedra

The descriptions of 3-D vector data histograms mentioned in previous reports are only an instance of a general class of polyhedra for which unusually quick solutions exist to the hidden line/surface problem. In the last six months, the conditions guaranteeing quick displayability have become understood, and display programs written to use the resulting algorithms [Brown 1978]. Also recently the original statistical motivation for the work has received more attention [Wellner 1978].

5. Component Building

5.1. Hardware

The Grinnell GMR-26 display device is on site and DMA-interfaced to the second (Vision) Eclipse computer. 32K of core has been added to the Vision Eclipse, which is also used for research in distributed computing (see Section 5.2). The original 80MB disk has been replaced with a 300MB one, and another 300MB disk has also been installed along with a much faster controller, leading to greatly enhanced performance. We are acquiring terminals and investigating how to meet our everyday computing needs by commercial, home-built, or combination intelligent terminal systems. Acquisition of a frame-rate TV-based digitizing device is still proceeding. The fast (50KB) link to the PDP-KL10 has been completed and is operating well.

5.2. Software

Advanced system software support is now used routinely, and more is under development. Communications protocols and distributed computing packages [Rovner 1978, Feldman 1978, Sheininger and Sabbah 1978, Selfridge 1978, Sloan 1978] have been developed to allow access to the GMR-26 through the local ALTO computers or the remote PDP-10, to achieve reliable transmission between distributed processes, to produce graphics and halftone images on ALTO screens from the PDP-10, and to allow file transfer and telnet to the Arpanet. The IPCF in the TOPS-10 operating system is the basis for communication between PDP-10 jobs, and these jobs may now create RIG messages and send them to the local operating system for disposition. At Rochester, the RIG message is the lingua franca that allows processes on remote machines to command the GMR-26, perform file manipulations, and other operations. Some of our work has been utilized by other image understanding groups, most extensively at SRI. wrote systems code for the multiple process HAWKEYE system [Barrow et al. 1977]. Some student projects in our Computer Vision course are aimed at producing useful system software for vision, and the common departmental interest in distributed computing assures that new and co-operative efforts using the distributed computation and communications packages will be launched frequently. A comprehensive library of vision routines [Sloan 1977-78] has been developed, centralized, documented, and incorporated into the NEXUS system. They allow interactive users a wide range of image-processing and display (graphics, halftone, color and B&W TV) capabilities. The work in image protocol is described in more detail in [Sloan, 1978] in these proceedings.

6. Motion Understanding

Understanding motion pictures has always presented an unusually difficult problem to computer vision efforts. The compelling gestalt induced in humans by moving objects is not well understood, and so there is little leverage on the immediate problems resulting from the large mass of data in multi- frame images. We are hoping to make progress first on a pared-down version of the problem which nevertheless offers an interesting set of perceptual phenomena to model. The domain is multi- frame images of animal motion; initial research is being carried out on sequential images of points of light attached to joints. This data can give humans a strong perception of coherent motion, and present work is aimed at understanding how we correctly identify points (about 13 in all in present data) from frame to frame, and how we segment the resulting moving points into meaningful body parts. Ultimately, the results will be applied to multi-frame grey-scale images. Data presently comes from a program which simulates a range of human walking motion in 3-D. The program is a useful theoretical tool, since it allows direct access (not mediated by vision) to movement parameters, and point locations. It is also a useful psychological research tool, since with it one can inexpensively investigate limits in human performance.

7. Texture

Textural areas can be thought of as those parts of an image where segmentation based on normal similarity measures fails. Meaningful analysis of textured areas must include discrimination between different textures and detection of parts of the same texture. The similarity of textures which are identical except for a scale change, a rotation, or a different range of intensities must be recognized.

We approach the texture problem by dividing texture regions into meaningful sub-elements of similar intensity sample points, then using rotation- and scale-invariant shape measures to characterize these regions and finally determining spatial relationships among our sub-elements. By using a decision tree program structure, easily discriminated textures are separated quickly, and more complex textural structure is extracted only when necessary [Maleson, 1978].

8. Programming Language Development

The Smart Compiler and Distributed Computation research groups are cooperating on a language for research into both these fields [Ball 1978]. It will contain the ideas of PLITS, together with improvements and extensions gleaned from the SAIL-PLITS implementations of the past. There are several separate ways in which the programming language developments are affecting Image Understanding research in our laboratory and elsewhere [Feldman & Williams 1977]. An overview of this work was presented at the last workshop [Feldman 1978]. Many of the ideas developed in this work are being heavily used in image understanding.

REFERENCES

Ball, J.E., et al., ZENO: a language for smart compiler research, Internal Memo, Computer Science Department, University of Rochester, 1978.

Ballard, D.H., Model-directed detection of ribs in chest radiographs, TR11, Computer Science Department, University of Rochester, March 1978.

Barrow, H.G., et al., Interactive aids for cartography and photo interpretation, Semiannual Technical Report, Artificial Intelligence Center, SRI International, November 1977.

Brown, C.M., Fast display of certain museum-viewable polyhedra, TR23, Computer Science Department, University of Rochester, March 1978.

Feldman, J.A., Synchronizing distant cooperating processes, TR26, Computer Science Department, University of Rochester, October, 1977.

Feldman, J.A., Systems support for advanced image understanding. DARPA Semiannual Technical Report, May 1978.

Feldman, J.A., and Williams, G. Some comments on datatypes, TR28, Computer Science Department, University of Rochester, December 1977.

Lantz, K.A., Procedural knowledge and control in a model - driven vision system, Thesis proposal, University of Rochester, February 1978.

Lantz, K.A., Ballard, D.H., and Brown, C.M. General invocation through procedure descriptions: two applications in image analysis, 22nd International Symposium of the Society of Photo-optical Instrumentation Engineers, San Diego, CA., August 1978.

Rashid, R.F. Motion understanding, Thesis proposal, University of Rochester, in preparation 1978.

Rovner, P.D. Flow control and reliable transmission in a system for distributed computing, TR22, Computer Science Department, University of Rochester, October 1977.

Rovner, P.D. Automatic representation selection for associative data structures, to appear in Proceedings of the National Computer Conference, Anaheim, CA., June 1978.

Russell, D.F., and C.M. Brown, Representing and using locational constraints in aerial imagery, Image Understanding Workshop, November, 1978.

Sabbah, D. Image calibration, Internal Memo, Computer Science Department, University of Rochester, in preparation 1978.

Scheininger, U., and Sabbah, D., The display process, Internal Memo, Computer Science Department , University of Rochester, December 1977.

Schudy, R., A model for echocardiography, TR 12, Computer Science Department, University of Rochester, (in preparation) 1978.

Selfridge, P., Name - value pairs in the Rochester vision header. Internal Memo, Computer Science Department, University of Rochester, January 1978.

Sloan, Jr., K.R., Rochester vision library documentation, Internal Memos, Computer Science Department, University of Rochester, 1977 - 1978.

Sloan, Jr., K.R., Effective transmission of raster images, Image Understanding Workshop, November, 1978.

Wellner, J.A., Two-sample tests for a class of distributions on the sphere, submitted for publication, February 1978.

SPATIAL UNDERSTANDING

T.O.Binford

Artificial Intelligence Laboratory, Computer Science Department
Stanford University, Stanford, California 94305

Abstract

The program is based on a model-based vision system, ACRONYM. It is integrated with research aimed at using local models in powerful stereo vision systems. ACRONYM incorporates a high level geometric modeling language which serves as an interface to the user. It uses a rule-based backward-chaining inference system for symbolic prediction of object appearances. It also includes a relaxation graph matching component which uses a coarse-to-fine strategy to interpret observed scenes.

Introduction

The objective of our research is to design and build a vision system which can accomplish typical tasks in photointerpretation and guidance. How the system does these tasks is as important as the fact that it does them. The system should be generalizable; also, it should enable an interpreter to specify tasks in a simple and natural way. The objective is approached by carrying out sample PI tasks in systems which will be assembled from a core of common modules integrated into a single system plus a few modules which are specific to the task. The tasks chosen include monitoring airfields and buildings, and locating airfields, aircraft, and vehicles in aerial photos.

Achieving the objective is not primarily a system effort. We must solve scientific problems whose solutions will lead to implementing algorithms which are crucial for carrying out these tasks. Some problems follow. 1. *An interpreter naturally specifies PI tasks in terms of object models, in terms of examples, and in terms of geometric relations.* In our approach, a high level modeling language functions as a convenient common language for the user and the system. Innovations in geometric modeling support our implementation of the modeling language.
2. *An interpreter solves a puzzle by piecing together selected and multiple clues from current images, background information, and previous images. In doing so, he relies heavily on spatial interpretation from stereo imaging and shadows, and spatial knowledge about structures.* Integrating multiple cues within a single task is a key issue which raises technical questions. We are defining a hierarchy of geometric representations in order to combine information which ranges from image level to surface level to object level to contextual level. We are exploiting local geometric representation to extend stereo mapping capabilities and integrate stereo with the system.

3. *An interpreter performs a wide range of tasks. Tasks have widely different collateral information at the contextual level; they vary widely at the object level; because of varied viewpoint, illumination, sensor, weather, and obscuration and camouflage, they vary greatly at the image level.* For a single system to map this wide range of task elements onto a common set of modules, it is convenient that the modules represent a natural decomposition of the problem into physically meaningful elements, for example, those we use in our own description of the problem. *It is important that the system be generic with respect to objects and generic with respect to viewing conditions. Our approach to generic interpretation is to use object models* made from generic parts, and to use symbolic prediction of appearances of objects, combined with descriptions of appearances made of generic parts.

Ultimately, interpreters will be able to instruct systems in natural language. In some problem areas, the current state of natural language systems appears near that goal. If natural language systems were sufficiently capable now, they could only translate between natural language and a general programming language such as LISP or FORTRAN. There is no very high level language for vision. The ACRONYM system is intended as a bridge between natural language and standard programming languages. *The representation hierarchy of ACRONYM is the basis* for a Vision Language.

Current Status

Progress on ACRONYM is summarized in a paper in these proceedings [Brooks]. The system has the form shown in figure 1. The geometric modeling subsystem contains a high level modeling language. It produces an Object Graph and a Context Graph. The predictor and planner subsystem is based on a rule-based backward-chaining system, patterned after Mycin [Bennett]. The predictor and planner makes a display of the model for the benefit of the user, and makes an Observability Graph, with estimates of local tactics for ordering graph matching. The matching subsystem has a coarse-to-fine relaxation mechanism for graph matching to go from predictions in the Observability Graph to observations in the Edge Graph and Surface Graph.

Object representations are based on generalized cones [Binford]. Generalized cones were designed to enable generic descriptions by part/whole graphs of generic parts. Generalized cone representations of objects are very compact. Complex parts can be modeled nearly as simply as a cube; a complex object has a representation with about the same complexity as the product of the number of parts times the complexity of the cube representation. The dependency hierarchy provides a natural set of levels of detail in description. Generalized cones provide relational information which is not available in surface representations and which is used in symbolic predictions of appearances of objects. An interesting aspect of the representation is that surfaces and cross sections of generalized cones are represented as 2d specializations of generalized cones, called ribbons. They are closely related to ruled surfaces.

The Observability Graph is an important element in the. hierarchy of representations. The Observability Graph is a collection of predictions of object appearances and relations from the Context Graph and the Object Graph. It corresponds to generic and special case observables. Generic observables are those which are quasi-invariant with respect to members of an object class, or quasi-invariant with respect to viewing conditions. For example, all passenger aircraft have a long generalized cone as the fuselage (generic with respect to object class); most views of the fuselage appear as elongated ribbons (generic with respect to viewing class). The predictor and planner rely on mapping generalized cones to 2d generalized cones, along with other mappings.

The Interpretation Graph contains correspondences between the Observability Graph and the Observed Graphs, i.e. the Edge and Surface Graphs. It makes heavy use of mappings from ribbons of the Observability Graph to ribbons and edges of the Observed Graphs, and from surfaces of the Observability Graph to surfaces of the Observed Graphs. It also makes use of maps in the other direction, i.e. mapping observed ribbons to predicted ribbons and observed surfaces to predicted surfaces. Mappings run both ways at all levels; thus the system can be run both bottom-up and top-down.

ACRONYM has been debugged on a toy example extracted by hand from high altitude aerial photographs of an airfield. It is now being tested on a real example from aerial pictures of San Francisco airport. Results from our research in stereo [Arnold] will be used, in combination with edge maps from Nevatia and Babu [Nevatia].

Research Plans

During the near future, geometric modeling capabilities will be extended. A library of primitives will be implemented to simplify descriptions in the high level modeling language. An interactive geometric editor like that of GEOMED [Baumgart] will aid the user; some ways of making the editor smart are being considered. New volume and surface primitives will be added; it was necessary to add another subclass of generalized cones to model a Lockheed L1011, and a few other modeling primitives will be useful for other tasks. Union, intersection, and difference operations are important for our representations and for the predictions which use the representations. Display with hidden surface elimination will be useful for user feedback. The compact representations enable efficient hidden surface suppression algorithms.

The predictor and planner will be extended in its use on the airport scene. Several more analytic solutions are necessary for the prediction of object appearance. We expect that the backward-chaining mechanism will prove useful initially, and that further testing will require new ways to accomplish prediction and evaluation of effectiveness of alternatives.

Our research on stereo mapping will be extended to integrate it with the ACRONYM system. First, it will segment and attach symbolic descriptors to surfaces in the image. Then, additional forms of local context will be used to improve its accuracy and robustness. Its performance appears now to be satisfactory for initial tests of ACRONYM and it appears capable of improvements which would satisfy requirements of subsequent tests of the system.

Collaboration

We are collaborating with Lockheed on a program of applying image Understanding concepts and techniques to mid-course guidance. The objective is to develop means for flexible flight path planning using passive visual sensing. Feasibility is based on devising ways of using reference images which require small storage requirements. A model for this program is a human navigator who uses a combination of inputs from several sensors, coupled with flying by landmarks. An approach is to use an integration of the input of sensors based on modeling of corrections to predicted flight path. Our effort has two parts. The firsrt part consists of evaluating the stereo ranging capabilities of Gennery's programs [Gennery] for the determination of altitude. The second is the further development of curve matching algorithms for navigating by landmarks, based on previous work of Bolles [Bolles]. The storage requirements for navigating by tracking linear landmarks are small.

References

Arnold, R. David [1978]: *Local Context in Matching Edges for Stereo Vision*, Proceedings: ARPA Image Understanding Workshop, Cambridge, Mass, May, pp. 65-72.

Bennett, J.S., L.G. Creary, R. Engelmore and R. Melosh [1978]: *A Knowledge-based Consultant for Structural Analysis*, Forthcoming Stanford CS Report, Nov.

Baumgart, Bruce G. [1974]: *Geometric Modeling for Computer Vision*, Stanford Artificial Intelligence Laboratory, Memo AIM-249, Oct.

Binford, Thomas O. [1971]: *Visual Perception by Computer*, Invited paper at IEEE Systems Science and Cybernetics Conference, Miami, Dec.

Bolles, Robert C. [1976]: *Verification Vision Within a Programmable Assembly System*, Stanford Artificial Intelligence Lab Memo AIM-295.

Brooks, Rodney A., Russell Greiner and Thomas O. Binford [1978]: *Progress Report on a Model-Based Vision System*, these Proceedings: ARPA Image Understanding Workshop, Carnegie-Mellon, Nov.

Gennery, Donald B. [1977], *A Stereo Vision System*, Proceedings ARPA Image Understanding Workshop, Stanford University, October 1977.

Nevatia, Ramakant and Ramesh Babu [1978]: these Proceedings: ARPA Image Understanding Workshop, Carnegie-Mellon, Nov, 1978.

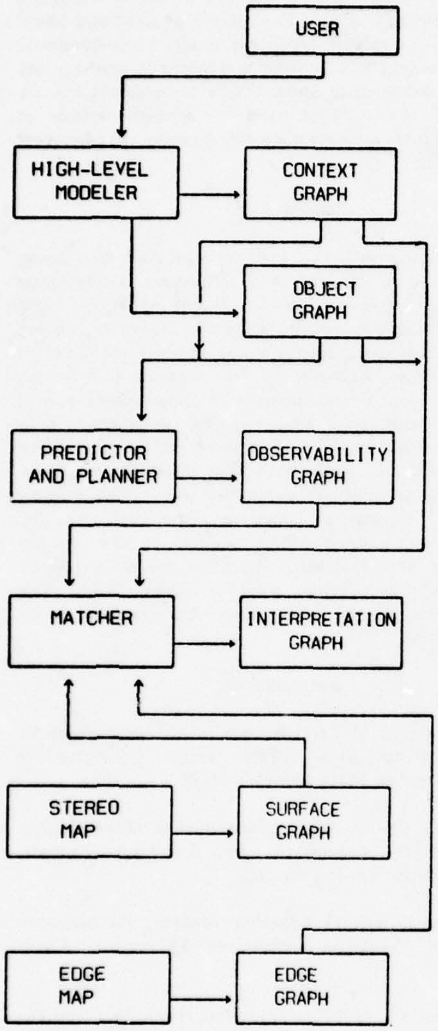


Figure 1: ACRONYM: Model-Based PI System

THE SRI ROAD EXPERT: AN OVERVIEW

M.A. Fischler (Principal Investigator),
G.J. Agin, H.G. Barrow, R.C. Bolles,
L.H. Quam, J.M. Tenenbaum, H.C. Wolf
SRI International
Menlo Park, California

ABSTRACT

This paper presents an overview of SRI International's on-going effort to construct a "Road Expert" whose purpose is to monitor and interpret road events in aerial imagery. Goals, approach, and the current state of this research are described.

INTRODUCTION

Image Understanding research at SRI International was initiated in 1975 to investigate ways in which diverse sources of knowledge might be brought to bear on the problem of analyzing and interpreting images. The initial phase of research was exploratory in nature, and identified various means for exploiting knowledge in processing aerial photographs for such military applications as cartography, intelligence, weapon guidance, and targeting. A key concept is the use of a generalized digital map to guide the process of image analysis.

The results of this earlier work were integrated in an interactive computer system called "Hawkeye" (see Ref 1). Research has now focused on a specific task domain: road monitoring. The following sections of this report present an overview of this on-going effort.

OBJECTIVE

The primary objective in this research is to build a computer system which "understands" the nature of roads and road events. It should be capable of performing such tasks as:

(a) Finding roads in aerial imagery

(b) Distinguishing vehicles on roads from shadows, signposts, road markings, etc.

(c) Comparing multiple images and symbolic information pertaining to the same road segment, and deciding if significant changes have occurred.

It should be capable of performing the above tasks even when the roads are partially occluded by clouds or terrain features, or viewed from arbitrary angles and distances, or pass through a variety of terrains.

APPROACH

To achieve the above capabilities, we are developing two "expert" subsystems: the "Road Expert" and the "Vehicle Expert". The Road Expert knows mainly about roads, how to find them (in imagery) and what things belong on them. It works at low to intermediate resolution (say from 1 to 20 feet of ground distance per image pixel) and has the ability to distinguish vehicles from other road detail. The Vehicle Expert works on higher resolution imagery and can identify vehicles as to type. We are concentrating our efforts on the Road Expert, and therefore will limit our discussion to this component of our system.

The major tasks (automatically) performed by the Road Expert are:

(1) Image/Map Correspondence: Place a newly acquired image into geographic correspondence with the map data base.

(2) Road Tracking: Precisely mark the centerline of selected visible sections of road in the image.

(3) Anomaly Analysis: Locate and analyze anomalous objects on, and adjacent to, the road surface; identify potential vehicles.

The image/map correspondence task is being accomplished primarily by using roads and road features as landmarks. Correspondence is performed at resolutions as coarse as 20 feet/pixel so that a reasonably wide field of view (10 to 100 square miles) can be processed at one time. Working iteratively to refine the position estimate and verify the detected features, this task deals with image detail over a 20:1 range of resolutions. It is nominally assumed that the ground location of the image is known to within +/- 200 feet.

Having placed the image into correspondence with our map data base, one or more of the visible road sections is selected for monitoring. The road center-line and lane boundaries are found to an accuracy of 1 to 2 pixels in imagery with a resolution of 1 to 3 feet/pixel.

Given the precise road locations in the image, anomalous objects are detected by scanning on and along the road pavement. These anomalous objects are then identified as to type (e.g., vehicle, shadow, road surface marking, signpost, etc.).

The above tasks are supported by information about road condition and general structure from a symbolic data base. For example, if prior photographic coverage of the area being analyzed is available, the problem of anomaly classification can be simplified by determining if a similarily shaped anomaly could be found in the same general location over some extended period of time. Additional examples of how data base knowledge and stored models can aid in the analysis process include: the use of time of day in discriminating shadows from objects of interest; the general shape and width of the road (as obtained from a map) to aid in road tracking; and the expected size, shape, and road orientation of potential vehicles.

A central theme of this effort is to consider roads as a knowledge domain. In particular, we are addressing the question of how a-priori knowledge can be directly invoked by the image analysis modules (what type of knowledge; how should it be represented; what are mechanisms for its use). To achieve our goal of building a very high performance system, we are developing explicit models of the image structures we are dealing with; and additionally, models of the decision procedures embedded in the image processing algorithms so that the algorithms can evaluate their own performance. Finally, we are planning an over-all control structure which will be concerned with the problems of coordinating analysis across a spectrum of levels of resolution, and with integrating multisource information.

PROGRESS

(1) Data Base Construction: An underlying assumption of our overall approach is the existence of a map data base to guide the image analysis process. A significant part of our effort is thus concerned with the questions of what information this data base should contain and how it should be structured, as well as with assembling the needed data.

We have selected five distinct geographic sites scattered around the San Francisco Bay Area, have acquired multiple photographic coverage for each of these sites, and are currently building a detailed data base for one of these sites (PM280). Figure 1 shows one of our images of this site, and Table 1 lists some of the entities that are included in the data base.

In addition to expanding the size and scope of our data base along the lines indicated above, we plan to use the capabilities of the Road Expert itself to automate many of the steps required for such data-base construction.

(2) Image/Data-Base Correspondence: This task involves locating a few known road features (landmarks) in a newly acquired image, and then using the correspondence between the location of these landmarks and their geographic coordinates as stored in our map data base to determine the precise location and orientation of the "camera" when the image was acquired. Given the camera parameters and a terrain map, we can now derive a transformation that will assign geographic (x, y, z) coordinates to every point in the image. Figure 2 shows some of the landmarks we are currently using for the PM280 site. The search in the image for the landmarks is a sequential process guided by our continually more precise estimate of the camera's location; as each landmark is found, we update the camera model to further reduce the search area required to locate additional landmarks. Figure 3 shows an example of the uncertainty ellipse generated by the "camera calibration strategist" to delimit the search for the first landmark. (This ellipse is based on a mathematical model of the calibration process and assumed a-priori knowledge of initial uncertainty in camera location.) Once the first landmark has been located, the camera calibration strategist can refine the position estimate and even further narrow the search for the second landmark as also shown in Figure 3.

Our work on the correspondence problem, employing an iterative approach which combines error modeling, feature matching, verification, and refinement of the camera location estimate, has resulted in a number of extensions to the existing theory. A more complete exposition of the above approach and its status is contained in a companion paper (Bolles et al., Ref. 3). However, it is important to note here that we have been able to automatically establish image/map correspondence to an average error of between 2 and 3 feet of ground distance. Thus, given the potential robustness of this approach, we believe that it can play an important role in an image-matching navigation or terminal homing system (e.g., the cruise missile).

Additional work on this particular task will be primarily directed to improving the performance and flexibility of our landmark detectors, especially in regards to the question of verification and filtering out of false matches.

(3) Road Tracking: We have developed a number of techniques capable of tracking roads in aerial imagery across a 1 to 20 feet/pixel spectrum of resolutions. These results have been described in previous reports (see References 1 and 2) and, under the conditions available in our current imagery, perform extremely well. Figure 4 shows the performance of the low resolution road tracker. The low resolution road tracker uses a road model which assumes local homogeneity in intensity along the road; it also assumes contrast in intensity between the road and the adjacent

terrain. A linking algorithm uses an optimization technique to find a "best estimate" of the global road path based on local agreement with the road model described above.

Figures 5a and 5b show some examples of the high resolution road tracker. Using a road model that assumes segments exhibiting relatively smooth/slow changes in direction and also in the intensity profile normal to road direction, we have been able to achieve surprisingly robust performance in tracking the road center line. In many cases, roads that have almost no discernible contrast at their edges can be readily followed. The way in which road tracking interacts with (and takes advantage of) the calibration process is described in Bolles et al. (Reference 3). Note that the clouds appearing in these images were artificially generated by a synthesis program we were forced to resort to in order to get a variety of cloud cover conditions needed to adequately test our techniques.

Future work on road tracking will be concerned with the problem of "verification" and with maintaining current levels of performance as the viewing conditions become increasingly more difficult (e.g., greater degrees of cloud cover or occlusion by shadows and adjacent terrain features). Rather than just making a best estimate of road location, we want the road tracker also to estimate the likelihood that this best estimate is indeed a visible segment of road.

(4) Anomaly Analysis: One method we are currently developing for detecting anomalous objects on the road surface is based on obtaining a local model of the variations in road reflectance and noting any significant deviations from this model. Figures 6a through 6d show the anomalies which were detected on a section of road using this approach. It would appear that vehicles can be distinguished from other anomalies, not only by their size and shape characteristics, but also by the fact that they have a range of local intensity variations (due to shadow, highlights from metal and glass, differently oriented surfaces, etc.) far exceeding most other road artifacts.

Our work on anomaly analysis is expected to receive a significant increase in attention over the next few months. In this context, shadow understanding, data-base information, and previous photographic coverage will be employed to help interpret detected anomalies.

CONCLUDING COMMENTS

We see the military relevance of our work extending well beyond the specific road monitoring scenario presented above. In particular, a Road Expert can be applied to such problems as:

(1) Intelligence: monitoring roads for movement of military forces

(2) Weapon Guidance: use of roads as landmarks for "map-matching" systems

(3) Targeting: detection of vehicles for interdiction of road traffic

(4) Cartography: compilation and updating of maps with respect to roads and other linear features (especially those concerned with transportation), such as airport runways, railroads, rivers, etc.

In accord with our generalized view of the applicability of the Road Expert and the knowledge-based image analysis techniques we are constructing, we are attempting to achieve a level of performance and understanding in each of the functional tasks which far exceeds that which would be required for dealing with the road monitoring scenario alone.

REFERENCES

1. H.G. Barrow et. al., "Interactive Aids for Cartography and Photo Interpretation: Progress Report, October 1977," Proceedings: Image Understanding Workshop, pp. 111-127 (October 1977).

2. L. Quam, "Road Tracking and Anomaly Detection," Proceedings: Image Understanding Workshop, pp. 51-55 (May 1978).

3. R.C. Bolles, L.H. Quam, M.A. Fischler, and H.C. Wolf, "The SRI Road Expert: Image-to-Database Correspondence," Proceedings: Image Understanding Workshop (in press, November 1978).

TABLE 1: ROAD EXPERT DATA BASE CONTENT

(1) Digitized Imagery (and a description of the acquisition process as well as the imaging parameters)

(2) Analyzed Images (results accompanied by processing history)

(3) Image Descriptions (manually annotated images; overlays; pointers to generic models of image objects, etc.)

(4) Predicted Images (under specified viewing conditions)

(5) Calibration Matrices (and associated landmarks and error estimates)

(6) Ground Truth (i.e., precise locations and dimensions of selected scene objects)

(7) Photometric and Geometric Models of data base objects (with pointers to image examples)

(8) Performance Models of Image Operators

(9) Corresponding image subsets from overlapping coverage of the same geographic area (preferably acquired automatically from the known calibration data associated with the images)

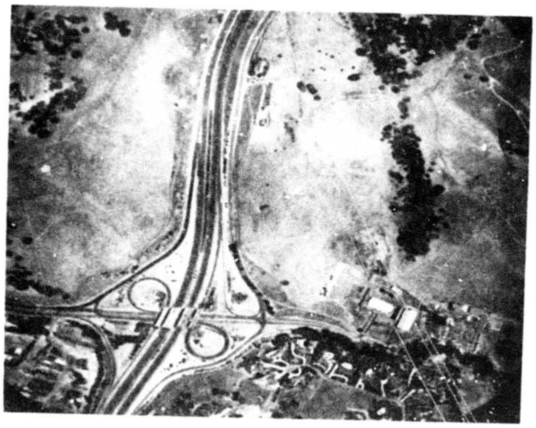


FIGURE 1 OVERVIEW OF THE PM280 SITE

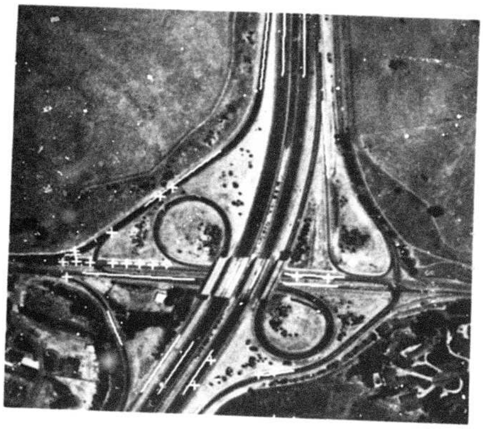


FIGURE 2(a) LOCATION OF PM280 SITE LANDMARKS

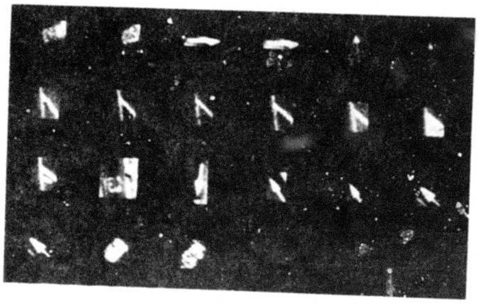


FIGURE 2(b) ROAD SURFACE MARKINGS USED AS "POINT" LANDMARKS

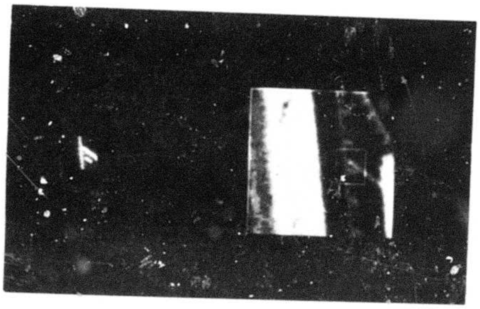


FIGURE 2(c) A POINT LANDMARK AND ITS APPEARANCE IN AN IMAGE

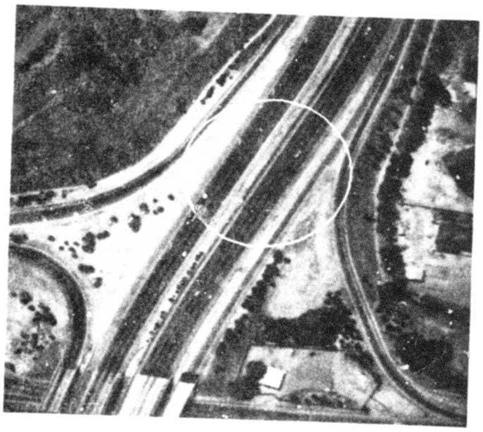


FIGURE 3 UNCERTAINTY ELLIPSES FOR LOCATING A KNOWN LANDMARK

The Larger Ellipse Represents the Initial Uncertainty in Locating a Road Surface Landmark. The Small Ellipse is the Refined Estimate of Location after One Other Nearby Landmark Has Been Located.

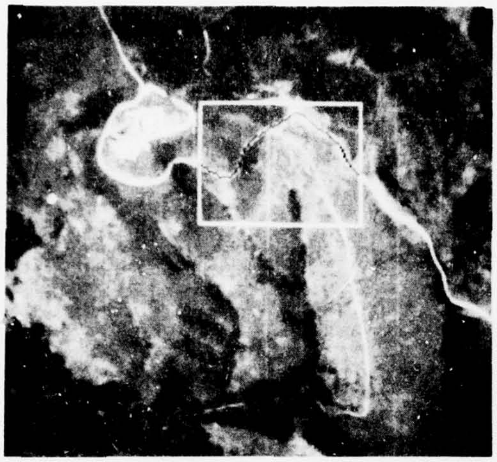


FIGURE 4 A ROAD LOCATED AND MARKED IN A SPECIFIED SEARCH WINDOW BY THE LOW RESOLUTION ROAD TRACKER

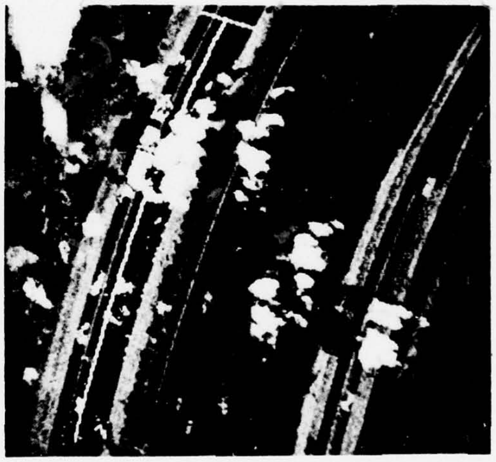


FIGURE 5(a) THE HIGH RESOLUTION ROAD TRACKER FOLLOWING A ROAD IN THE PRESENCE OF CLOUD COVER

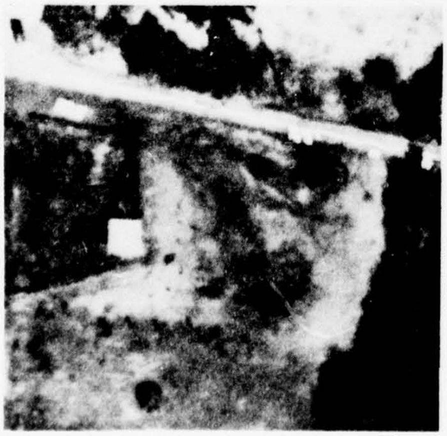


FIGURE 5(b) THE HIGH RESOLUTION ROAD TRACKER FOLLOWING A DIRT ROAD

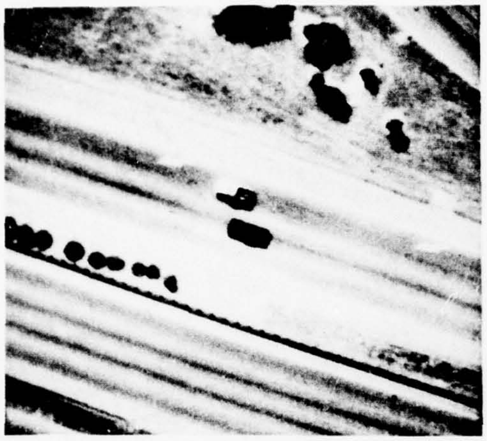


FIGURE 6(a) ORIGINAL SEGMENT OF AN IMAGE

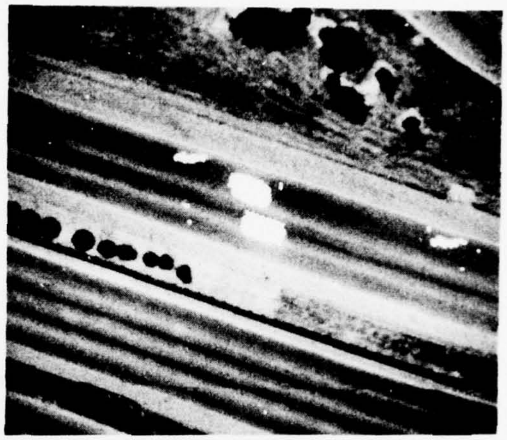


FIGURE 6(b) DETECTION OF ANOMALOUS AREAS ON THE ROAD SURFACE

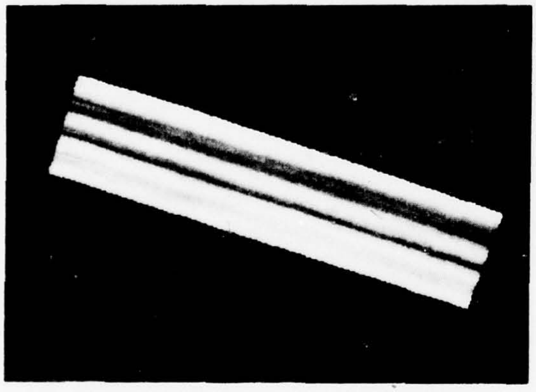


FIGURE 6(c) INTENSITY MODEL OF THE ROAD SURFACE

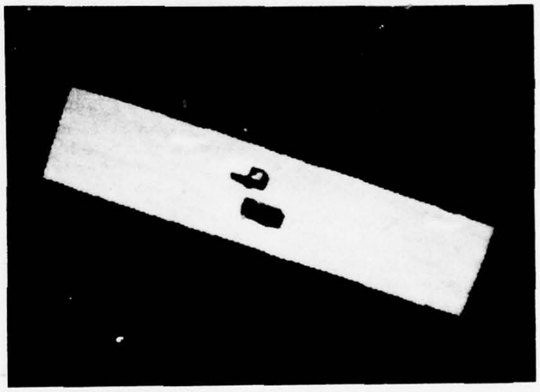


FIGURE 6(d) SUBTRACTION OF NOMINAL ROAD SURFACE INTENSITIES TO ENHANCE ANOMALIES FOR FURTHER ANALYSIS

USC OVERVIEW AND IMAGE UNDERSTANDING DEMONSTRATION UNIT

By

Harry C. Andrews

Image Processing Institute
University of Southern California, Los Angeles, California 90007

RESEARCH OVERVIEW

This document represents the results of research developed over the past 6 months at the USC Image Processing Institute. Research has been devoted to 3 major areas: image understanding, image processing, and smart sensor design. These areas are abstracted below.

IMAGE UNDERSTANDING PROJECTS

The image understanding tasks presented in this semiannual report are focused in first level and second or higher level processing procedures. In the first level processes, edge and texture techniques are developed. Edge analysis results are presented in which quantitative measures of performance on a variety of different edge operators are evaluated. Different performance functions, such as edge detection, positional accuracy, invariance of operator to orientation, etc., are utilized. In the area of texture work both analysis and synthesis procedures are reported. Texture analysis via optical filtering and the use of color representation has been demonstrated to be an effective means of detection and visualization of specific texture patterns. In the synthesis of texture a stochastic whitening process is developed which looks extremely hopeful as a tool in defining features for texture recognition and discrimination. Another texture synthesis technique is presented which is based upon the statistical (N-gram) approach. This method although still in its one-dimensional form, show promise in its avoidance of moment techniques. Finally, some novel "segmented window" first layer processing techniques are presented with hypotheses as to their usefulness in ongoing research.

In the arena of second or higher level processors, feature usages of small Fourier transforms on reflectance imagery, edges, direction of edges and density of edges is developed. Edge detection, linking, and line finding algorithms as well as descriptions of linear segmented objects are presented as work in progress for various image segmentation scenarios. Finally, higher level operating software principles are formulated and examples of data structures and their relationships are presented.

IMAGE PROCESSING PROJECTS

A variety of image processing projects are reported herein. They fall into three general areas of computational procedures, restoration methodologies, and inverse SAR imaging. A presentation is made on the computation of the condition number of a matrix to predict the degree of ill-conditioning and subsequent potential degrees of freedom in such a process. Such computations become extremely useful for large matrix processes as found in most imaging applications. In the generation of computer hologram interpolations, a special computational savings is developed to avoid the inefficiencies of zero padding traditionally used in most Fourier image filtering techniques.

In the arena of image restoration two techniques are reported upon. Results from the method of blind a posteriori restoration are presented in pictorial form. A new method of Poisson MAP restoration is also developed and analysis presented in which improved sensor models for imaging result.

Finally, two papers on inverse synthetic aperture radar imaging are presented. One is formative in is presentation and proposes to image shadowed regions via RATSCAT turntable data. The second represents processing results from an inflight aircraft in both a straight flight and a turn set of geometries. Resulting imagery is presented.

SMART SENSOR PROJECTS

The following report from Hughes Research Laboratories reflects the continuing progress on the CCD smart sensor design front. As usual we are pleased to see such results and wish to point out that this represents a classic illustration of technology transfer as the US Army NVL has contracted and received one of our earlier circuit chips in an operating unit. Recent chip design will afford 7x7 processing as well as programmable arrays and limited feature selection in our ultimate effort for the computation of a texture CCD circuit.

RECENT GRADUATES

One of the Image Processing Institute's most precious products is its graduate students and it is always a pleasure to see our students graduate and move on to professional positions. This section lists the abstracts of the dissertations of the three most recent graduates and represents research in edge detection, restoration, and radar imaging. We are proud of their work and wish them well in their endeavors. Details of their disserations appear as USCIPI technical reports and are available upon request for those interested.

RECENT PUBLICATIONS

The report closes with a listing of Institute research staff publications. The majority of these are in the reviewed open literature and are an indication of the health of our research ideas. Naturally, due to the review process a delay in published results occurs for the open literature publications.

TABLE OF CONTENTS

To further provide the reader with insight as to who is working on what research tasks at USC, the table of contents of our upcoming semiannual report is listed below.

DEMONSTRATION UNIT

The Image Processing Institute at USC is configuring an exploitation station to eventually be installed at ARPA headquarters in Arlington, Virginia. This unit will be both an ARPANET terminal and will also be a stand alone station, both configurations of which will allow on line and off line real time demonstrations of many of the IU contractors' results. The station will consist of the following items:

PDP 11/34
2 Disks
1 Tape Unit
1 Terminal
1 Comtal Vision I Display
1 ARPANET Interface

In anticipation of the possible use of this unit as a facility for the "demonstration phase" of ARPA's IU program, USC is soliciting opinions from the IU contractor community as to desireable software and hardware interfaces. Of paarticular interest is the ARPANET transfer mode you wish to use and the optimal use of the RAM memory on the COMTAL unit for both graphics, color imagery, and roaming capability. Mr. Toyone Mayeda is in charge of this project and any inputs should be directed to him at USC IPI.

IMAGE UNDERSTANDING AND INFORMATION EXTRACTION

T. S. Huang
K. S. Fu

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

I. OVERVIEW

The objective of our research is to achieve better understanding of image structure and to improve the capability of image processing systems to extract information from imagery and to convey that information in a useful form. The results of this research are expected to provide the basis for technology development relevant to military applications of machine extraction of information from aircraft and satellite imagery.

The main themes of our research are:

1) To find good symbolic representations for images.
2) To develop techniques for transforming raw image data into such representations.

Symbolic representations under consideration include relational graphs, syntactic methods, and syntactic-semantic methods. The process of transforming raw image data into symbolic representation is a complex one; therefore, we subdivide it into several steps as shown in Fig. 1. We first consider the left side of the block diagram in Fig. 1. After the sensor collects the image data, the preprocessor may either compress it for storage or transmission or it may attempt to put the data into a form more suitable for analysis. Image segmentation may simply involve locating objects in the image or, for complex scenes, determination of characteristically different regions. Each of the objects or regions is categorized by the classifier which may use either classical decision-theoretic methods or the more recently developed syntactic methods. In linguistic terminology, the regions (objects) are primitives, and the classifier finds attributes for these primitives. Finally, the structural analyzer attempts to determine the spatial, spectral, and/or temporal relationships among the classified primitives. The output of the "Structure Analysis" block will be a description (qualitative as well as quantitative) of the original scene. Notice that the various blocks in the system are highly interactive. Usually, in analyzing a scene one has to go back and forth through the system several times.

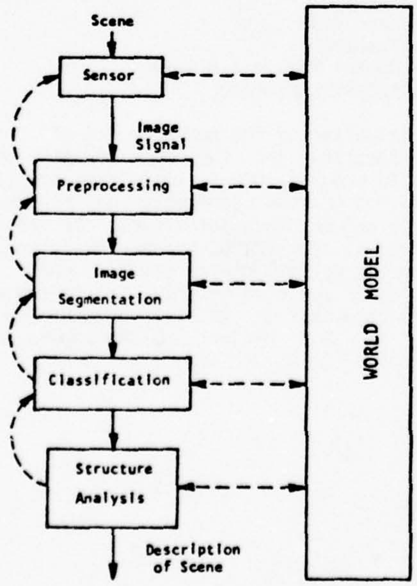


Fig. 1 An Image Understanding System

Past research in image understanding and related areas at both Purdue and elsewhere has indicated that scene analysis can be successful only if we restrict a priori the class of scenes we are analyzing. This is reflected in the right side of the block diagram in Fig. 1. A world model is postulated for the class of scenes at hand. This model is then used to guide each stage of the analyzing system. The results of each processing stage can be used in turn to refine the world model.

Before we start to analyze a scene, a world model is constructed which incorporates as much a priori information about the scene as possible. This could, for example, be in the form of a relational graph containing unknown parameters. Then the analysis problem becomes the determination of these unknown parameters. In this way, the difficult problem of scene analysis is reduced to the (conceptually) much simpler problems of detection, recognition, and mensuration.

Our research projects fall into the following overlapping categories: Preprocessing, Image Segmentation, Image Attributes, Classification Techniques, Image Structure, Applications, and Implementation.

II. SUMMARY OF RESEARCH PROJECTS

A. Preprocessing

(a) Image restoration. We are studying both theoretically and by computer simulation the behavior of an iterative method of restoring images degraded by linear shift-varying systems which we developed several years ago [1]. This method is computationally much more efficient than other methods such as singular value decomposition. However, for large images (1024x1024 points), it is still far too time consuming. We are looking into the use of array algebra [2] to speed up the process.

(b) Image enhancement. We have initiated a basic research project in nonlinear image enhancement techniques. Of particular interest is the problem of reducing noise in images without blurring the sharp edges contained therein. Our approach is to decompose the image into several components in such a way that the noise characteristics in the components are more amenable to nonlinear filtering methods. One particular class of nonlinear techniques under study is median filtering and its extensions. A fast two-dimensional median filtering algorithm has been developed and programmed on our PDP 11/45 computer. It is several orders of magnitude faster than the most efficient sorting methods [3].

(c) Image coding. A spatial-domain efficient coding method has been developed [4] which is comparable in performance to transform coding but much easier to implement. We are currently collaborating with Rome Air Development Center in evaluating the effects of several coding methods on aerial photo image quality from the point of view of photo-interpreters.

(d) Registration. Registration is a key step in processing sequences of images. For example, averaging several successive image frames to reduce noise should be preceded by frame registration. We have developed a registration technique which can be implemented easily in real time. This scheme is suitable for applications involving a FLIR or a conventional TV system. Each image is converted into a binary feature image. Feature images may be rapidly registered and also any movements of significant objects within the image can be detected [5].

B. Segmentation

(a) Edge detection. An extensive study is being carried out on the use of statistical hypothesis testing in edge detection. Both parametric and nonparametric methods are being investigated. It has been found that the use of Wilcox's test is especially effective.

(b) Region growing. The original region-growing BLOB algorithm [6] for image segmentation processes the image in a sequential line by line fashion. This is inefficient, if we are looking for a particular type of segments (e.g., airplanes in an aerial photo). We have modified the algorithm so that it will follow the boundary of a segment from a given initial point.

(c) Clustering. For each image point, several features in a small neighborhood around it are measured. In this way, all the points in an image are mapped into a feature space. A clustering algorithm is then applied to the feature space. Finally, each cluster is mapped back to the image space to get the segments [7]. We have used a graph-theoretic clustering algorithm which has the advantage that the number of clusters does no have to be specified a priori, and are currently looking into suitable texture features for various applications.

C. Attributes

(a) Shape analysis. We have made significant improvement in the computational efficiency of Fourier boundary descriptors for shape analysis, and have extended the method to the recognition of three-dimensional objects. Extensive computer simulations have been carried out where the technique is applied successfully to the recognition of three-dimensional airplanes [8]. One disadvantage of the Fourier descriptors is that they are global properties of the boundary and therefore will not work if the boundary of an object is not completely obtained by segmentation because of background noise and interference. We are therefore starting a basic research project on the use of local shape descriptors to do recognition [9].

(b) Texture analysis. We have developed a simple texture measure, called the maxmin descriptors [10], which are very easy to compute yet perform as well as other much more complicated measures such as those based on spatial-dependence matrices. These maxmin texture descriptors are being applied to several problems in image segmentation and object recognition.

D. Classification

(a) Statistical classification using contextural information. Classification of multispectral image data is routinely carried out by classifying a single pixel at a time, extracting information from the spectral domain, ignoring the two-dimensional or image character of the data. Recent studies confirm that there is useful information in the context of a pixel (e.g., its neighbors) which can be helpful in identifying the pixel. In this research the scene is considered to be a multi-dimensional random process characterizable in terms of its statistical transition properties. Implementation of classification rules utilizing these properties without being prohibitively expensive in terms of computational requirements represents a considerable challenge. Two procedures have been developed for classification using context [11,12]. They were applied to LANDSAT data with considerable success--the classification error percentages in many cases were reduced by half (compared with pixel by pixel classification). We are at present planning to implement one of the procedures on a CDC Cyber-Ikon computer.

(b) Feature subset selection. In many classification problems, the number of potentially useful features is large. We face the problem of choosing among them a small subset in an optimum or nearly optimum way. an exhaustive search is time consuming. Based on the branch-and-bound approach, we have developed an efficient method of making the choice [13]. In the examples we tried, this method is 50 or more times faster than the exhaustive search method.

E. Structure Analysis

(a) Tree grammar. A two-dimensional grammar called tree grammar has been developed. It was used to characterize shape [14] as well as texture [15]. One major advantage of tree grammar is that its parsing is very similar to that of a string grammar.

(b) Combining symbolic and numerical descriptions of images. One disadvantage of the syntactic approach is that it is awkward for describing numerical properties of patterns. We have been involved in two research projects where a marriage of symbolic and numerical image descriptions is carried out. In the first project, attributed grammar is used for shape description [16]. In the second project, various semantic considerations are introduced into the production rules of a grammar [17]. Both approaches have been applied to airplane detection and recognition in aerial photographs with encouraging results.

F. Applications.

The various results from our basic research projects described above are being used to attack several mission-oriented problems.

One problem is real-time video tracking. We have just started this project, collaborating with the U. S. Army White Sand Missile Range. WSMR has supplied us 20 digitized video images--an additional 150 images are soon to be added.

Another problem, which we have made considerable progress on, is FLIR target detection and recognition. We have been working on this project together with Honeywell, who supplied us 120 FLIR images with identified tactical targets. We have developed successful algorithms for image segmentation and target recognition for FLIR images in a rural scenario [18]. We plan to look into the much more difficult urban scenario in the future.

G. Implementation.

Most military, industrial, and commercial image analysis applications require either real-time processing or a very large data base, or both. Therefore, efficient implementation of algorithms is of the utmost importance.

When we develop algorithms in our basic research projects, we pay special attention to implementation considerations. In addition, several implementation-oriented projects are being initiated. These include a study on computer architectures for image processing [19] and hardware implementation of a binary array processor.

III. Future Research Directions.

Our research objective and main themes remain unchanged. However, in the future, our emphasis will be turned more and more to the analysis of image sequences which contain motion or scene changes. Each block in Fig. 1 and the interrrelations among the blocks will be reexamined with image sequences in mind.

References

1. T.S. Huang, D. Baker, and S. Berger, Iterative image restoration, Applied Optics, May 1975.
2. V.A. Rauhala, Array algebra, Fotogrammetriska meddelanden, Vol. 6, No. 6, June 1974.
3. T.S. Huang, G. Yang, and G. Tang, A fast two-dimensional median filtering algorithm, in Proc. of IEEE Conference on Pattern Recognition and Image Processing, June 1978, Chicago.
4. O.R. Mitchell, E.J. Delp, and S.G. Carlton, Block truncation coding, in Conference Record of ICC 1978, Vol. 1, June 4-7, 1978, Toronto, Canada.
5. O.R. Mitchell, A.P. Reeves, et. al., Segmentation and classification of targets in FLIR and video imagery, in Proc. of Eighth Annual Symp. on automatic Imagery Pattern Recognition, April 3-4, 1978, NBC, Gaithersburg, Maryland.
6. J.N. Gupta and P.A. Wintz, A boundary finding algorithm and its applications, IEEE Trans. on Circuits and Systems, April 1975.
7. J. Yoo and T.S. Huang, Image segmentation by unsupervised clustering, Technical Report, TR-EE 78-19, May 1978, School of Electrical Engineering, Purdue University, West Lafayette, Indiana 47907.
8. T. Wallace and P.A. Wintz, Three-dimensional airplane shape recognition, to appear in IEEE Trans. on Computers.
9. T. Wallace, P.A. Wintz, and O.R. Mitchell, Advances in shape description with application to three-dimensional aircraft recognition, in this volume.
10. O.R. Mitchell and S.G. Carlton, Image segmentation using a local extrema texture measure, Pattern Recognition, Vol. 10, No. 3, 1978.
11. E.F. Kit and P.H. Swain, An approach to the use of statistical context in remote sensing data analysis, Proc. Fifth Canadian Symp. on Remote Sensing, Victoria, B.C., Canada, August 1978.
12. T.S. Yu and K.S. Fu, Contextual pattern classification for remotely sensed multispectral data, Proc. Eighth Modeling and Simulation Conference, Pittsburgh, PA, April 1977.
13. P.M. Narendra and K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Trans. on Computers, September 1977.
14. J. Keng, A syntactic method for image segmentation, Proc. Seventh Annual Symp. of automatic Imagery Pattern Recognition, Electronic Industrial Association, College Park, Maryland, May 23-24, 1977.
15. S.Y. Lu and K.S. Fu, A syntactic approach to texture analysis, Computer Graphics and Image Processing, June 1978.

16. K.C. You and K.S. Fu, Syntactic shape recognition using attributed grammars, Proc. Eighth Annual Symp. Automatic Imagery Pattern Recognition, April 3-4, 1978, NBS, Gaithersburg, Maryland.

17. G.Y. Tang and T.S. Huang, A syntactic-semantic approach to image understanding and creation, to appear in IEEE Trans. on Pattern Analysis and Machine Intelligence, April 1979.

18. O.R. Mitchell and S.M. Lutton, Segmentation and classification of targets in FLIR imagery, in this volume.

19. J. Keng and K.S. Fu, A special computer architecture for image processing, Proc. IEEE Conf. on Pattern Recognition and Image Processing, May 31-June 2, 1978, Chicago.

PROJECT STATUS REPORT
"IMAGE UNDERSTANDING USING OVERLAYS"
(Contract DAAG53-76C-0138)

Azriel Rosenfeld
Principal Investigator

Computer Vision Laboratory, Computer Science Center
University of Maryland, College Park, MD 20742

This project, initiated in April 1978, is a continuation of the project entitled "Algorithms and Hardware Technology for Image Recognition" (May 1976-March 1978). It is monitored by the U.S. Army Night Vision Laboratory, Fort Belvoir, VA; the project monitor is Dr. George Jones. The Westinghouse Systems Development Division, as a subcontractor, is investigating hardware implementation of the techniques being developed by Maryland; Dr. Glenn E. Tisdale is program manager for Westinghouse.

The earlier project [1] was concerned primarily with tactical target detection on forward-looking infrared (FLIR) imagery. Specific efforts involved image modeling, smoothing, noise cleaning, edge detection and thinning, thresholding, tracking, feature extraction, and classification. Through the use of convergent evidence, based on coincidences between edge maxima and borders of above-threshold regions, excellent object extraction performance was achieved. Westinghouse studied the CCD implementations of many of the algorithms that were developed, and breadboarded one basic function, a sorter. Communication among the Maryland, Westinghouse, and NVL groups was very good and led to greatly accelerated transfer of advanced image understanding techniques.

The present project is currently concerned with more complex infrared images containing several different types of objects (targets, trees, smoke plumes, markings on the ground, etc.). Several examples are shown in Figure 1. Interpretation of these images is quite difficult even for humans, and requires considerable use of contextual information. On the other hand, the number of object types and relationships among them is limited, so that the amount of processing required to understand these images should be manageable. Thus an immediate goal of the project is to demonstrate an image understanding capability for a class of real-world images having relatively simple descriptions.

The extraction and identification of regions in the images requires coordination of several types of information. Relaxation methods [2,3] should be useful in this connection. Comparison of several successive frames may also be necessary; flexible matching techniques, possibly involving relaxation, are under investigation for this purpose. Relaxation-like approaches will also be used in the initial image preprocessing and segmentation, as will methods based on the use of convergent evidence.

Work is currently being done on specific tools and techniques which are expected to become part of the overall system, or to contribute to its design. These areas are briefly summarized in the following paragraphs. Two of them are treated in greater detail elsewhere in these Proceedings [4,5]. The Westinghouse efforts will not be reviewed here; they are discussed in another paper in these Proceedings [6].

a. Data base acquisition. Several data sets have been acquired from NVL. They have been read from tape and prepared for subsequent processing by scaling and windowing. They are currently being viewed in order to identify, as reliably as possible, the objects that appear on them. The types of evidence used in these identifications have been tabulated and will serve as guidelines for the design of the image understanding system.

b. Image modelling. A class of image models based on random geometric processes is under investigation on an AFOSR grant [7]. It is planned to apply these models to infrared images, in order to statistically characterize the results of various processing operations applied to these images. For example, it should be possible to characterize the strengths of edge detection responses, which will be useful in defining thresholds for discriminating against noise responses. Thus successful modeling of a given class of images should provide a basis for the quantitative design of preprocessing and segmentation operations to be applied to these images.

c. Preprocessing. A number of specific preprocessing studies have been conducted. One of these is a comparative study of noise cleaning techniques, emphasizing iterative local (space-domain) operators [8]. An extension of this study to color or multispectral imagery is planned. A general software system for implementing and testing iterative local array operations is under development; it will be used to experiment with a variety of relaxation-like image preprocessing and segmentation operations.

d. Edge detection. A comparative study of multispectral edge detection techniques has been conducted [9]. An iterative approach to improving local estimates of edge magnitude and orientation has been implemented and applied to improving the detection of straight edge segments [10]. Further work on analyzing the straight edge content of images is planned.

e. Edge/border coincidence. On the previous project, a method of object extraction based on a combination of thresholding and edge detection was developed. Specifically, for any given threshold, let C be a connected component of above-threshold points; then we regard C as an "object" if most of its border points are local maxima with respect to edge value [11-13]. This method works well for isolated, "thresholdable" objects, but it breaks down for more complicated scenes, where the connected components do not correspond to single objects. An alternative approach in such cases is to use the thresholds to help select edge points, rather than using the edge points to select thresholds. In particular, edge points can be linked if they lie on a common border with respect to a given threshold, and the links can be strengthened if this is true for many thresholds. This approach is discussed in greater detail elsewhere in the Proceedings of this Workshop [4].

f. Pattern matching. Matching images of the same scene by array correlation is a computationally costly process, and is also sensitive to geometrical distortion and other types of systematic discrepancies between the images. One way to overcome this is to segment the images and match the resulting regions based on their global properties. Another possibility is to extract local features from the image and match the spatial patterns of these features. Good matches can be obtained provided these patterns have significant numbers of points located in approximately corresponding positions. Some experiments in point pattern matching are described elsewhere in these Proceedings [5]. The use of relaxation to define degrees of association between pairs of points in the two patterns is also under investigation [5].

g. Structure matching. Relaxation methods can also be used to match graph structures; local context can be used to define possible pairings of graph nodes. A few iterations of this process generally yields unambiguous pairings. More generally, in the case of weighted graphs, local context can be used to assign confidences to possible pairings; when this is iterated, the confidences of the correct matches remain high, while those of the incorrect pairings become very low. Experiments with the use of relaxation for graph matching are described in greater detail elsewhere in these Proceedings [5]. A general software system for implementing and tesing relaxation processes on graphs is under development. It will be used for experiments on region and object identification based on contextual information.

REFERENCES

1. Algorithms and Hardware Technology for Image Recognition, Final Report on Contract DAAG53-76C-0138, March 1978.

2. A. Rosenfeld, Relaxation methods: recent developments, Proc. Image Understanding Workshop, October 1977, 28-30.

3. A. Rosenfeld, Some recent results using relaxation-like processes, Proc. Image Understanding Workshop, May 1978, 100-104.

4. D. L. Milgram, Edge point linking using convergent evidence, in the Proceedings of this Workshop.

5. A. Rosenfeld, Some experiments on matching using relaxation, in the Proceedings of this Workshop.

6. T. J. Willett, Hardware implementation of image processing techniques, in the Proceedings of this Workshop.

7. N. Ahuja, Mosaic models for image analysis and synthesis, University of Maryland, Computer Science Center Technical Report 607, November 1977.

8. J. P. Davenport, A comparison of noise cleaning techniques, University of Maryland, Computer Science Center Technical Report 689, September 1978.

9. P. V. Sankar, Color edge detection: a comparative study, University of Maryland, Computer Science Center Technical Report 666, June 1978.

10. S. Peleg, Straight edge enhancement and mapping, University of Maryland, Computer Science Center Technical Report 694, September 1978.

11. D. L. Milgram, Region extraction using convergent evidence, Proc. Image Understanding Workshop, April 1977, 58-64.

12. D. L. Milgram, Progress report on segmentation using convergent evidence, Proc. Image Understanding Workshop, October 1977, 104-108.

13. D. L. Milgram, Region extraction using convergent evidence, University of Maryland, Computer Science Center Technical Report 674, June 1978.

IMAGE UNDERSTANDING RESEARCH AT CMU:
A Progress Report

Raj Reddy
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pa. 15213

INTRODUCTION

The primary objective of our research effort is to develop techniques and systems which will lead to successful demonstration of image understanding concepts over a wide variety of tasks, using all the available sources of knowledge. We are focusing our attention on three areas of research. First, we are developing an integrated concept demonstration of an image understanding system. The long-term goal of this research is to understand how knowledge can be used in the image interpretation process to produce systems which are 2 to 3 orders of magnitude more cost-effective than current systems. Over the next three years we expect to investigate how knowledge of maps, size and shape of landmarks such as buildings and rivers, and contextual relationships can be used in the interpretation of satellite images of the Washington, D.C. area and color scenes of downtown Pittsburgh.

The second area of research is the development and validation of concepts for computer architectures used in image understanding. The long-term objective of this research is to develop new computer architectures which will make low-cost image processing a serious possibility. We plan to evaluate the desirability of new processor designs and new instruction sets for image processing applications.

The third area is the development of intelligent interactive aids for tasks such as photo interpretation and map generation. Many of the same techniques which are useful in automatic interpretation are applicable in this area, except that in this case the human being provides the goal direction. The availability of intelligent assistants capable of examining large image data bases and retrieving desired information is expected to significantly improve human productivity in tasks such as photo interpretation and cartography.

The following is a brief summary of our work over the last six months.

KNOWLEDGE REPRESENTATION AND SEARCH

The ARGOS Image Understanding System (Rubin, 1978) has made some interesting advances since the last workshop. The system is now running with arbitrarily shaped segments instead of pixels. This makes it much faster, somewhat more accurate, smaller, and able to handle more knowledge sources. Current work is using hand-drawn segments, but a system using automatic segmentation using clustering (Ohlander, 1975; Price, 1976; Shafer, 1978) will soon be available.

Another investigation is the use of hierarchies of knowledge. To explore this, the City of Pittsburgh recognition task was divided into two sub-tasks: view angle identification and object identification. It is expected that the results of view angle task can help the object task to make overall recognition much more accurate. During this investigation, however, some interesting results have been obtained for the view angle identification task. To be able to perform this task, the system was trained with 24 machine-generated views of its internal model of the city at 15 degree increments around the center of the model. Each of the fifteen photographs of the city was then run against this knowledge base. In most cases, ARGOS pinpointed the view angle accurately. The average error was 30 degrees for training photographs, 51 degrees for test photographs.

IMAGE FEATURE ANALYSIS AND SEGMENTATION

In research reported elsewhere in this volume, Kender is exploring ways of deriving shape, orientation, and position information from textural gradients present in a scene. We hope to use such information, derived from static monocular images, as an additional source of knowledge for the downtown Pittsburgh task. The research has produced a new, general aggregation transform. In addition to being useful in perspective-related texture gradient work, it can also simplify, both conceptually and computationally, existing Hough-like transformations in domains dealing with vectored quantities: for example, line detection derived from edge vectors, or segmentations derived from motion-intensity gradient interaction.

We are continuing to study the effective use of knowledge in image segmentation. The KIWI segmentation program (Shafer and Kanade, in prep.) has incorporated a fast algorithm for extracting descriptions of regions resulting from a possible segmentation. By analyzing these descriptions, noise elimination can be performed without the use of global smoothing techniques. The speed of this process allows KIWI to examine, in parallel, several possible segmentations based on different image features, and to select the segmentation which results in the most viable region configuration.

KIWI provides a flexible framework for use in studying segmentation issues. Each decision made by the program can be manually inspected and overridden by the researcher; or, the system can be told to continue automatically until segmentation is complete. The specific operational programs may be selected at run-time by the experimenter; and the overall segmentation scheme may be redefined without interfering with the automatic record-keeping performed by the system. This allows maximum focus of attention upon particular aspects of the segmentation process, and the smooth integration and exploration of alternative techniques.

The KIWI system is currently operational, and is being used to provide automatically segmented image data for use by the ARGOS Image Understanding System (Rubin, 1978) in its experiments with arbitrarily shaped image segments.

3-D MODELING

Kanade is working on the problem of recovering 3-dimensional configurations of the scene from its image. The theory of Origami world (Kanade, 78) models the world as being made of surfaces, unlike conventional worlds, such as the trihedral world, which assume solid objects. Given a line drawing, the labeling procedure of the Origami world can recover the possible 3-D configurations which the drawing can have: not only it assigns line labels (convex, concave, occluding) as the Huffman-Clowes-Waltz scheme, but it recovers the relations on surface orientations much more systematically.

Application of the theory to real world images (chair scene) is now under progress. Color edge profiles taken across the edge are examined. Distances defined on the profiles are useful to tell what lines are similar to what lines. Geometrical properties, such as matched Ts, can provide plausible combinations of line labels. Heuristics concerning surface orientations, such as "parallel lines in the picture are usually also parallel in the scene", are also found very useful. All these knowledge can be nicely incorporated into the labeling procedure of the Origami world to obtain a unique or a few number of interpretations of the image. Also, how the results of labeling and relations among surface orientations thus obtained are used to obtain shape descriptions of the objects in the scene and to match them againt various concepts, say, "box", "chair", etc., is being studied.

INTERACTIVE AIDS

We are continuing with the development of the MIDAS database system (McKeown and Reddy, 1977) and are currently working to integrate map knowledge of the Washington, D.C. area into our system. The map knowledge consisted of a terrain (elevation) database and cultural features such as rivers, major buildings, forests and roads. We plan to apply this knowledge in a system which will match satellite and areal photographs to the terrain model and extract information from the images using the cultural feature data.

We have begun to investigate formalisms to define and extract terrain features (ridges, plateaus, hillsides, ravines etc.) given elevation data. These symbolic terrain feature descriptions will be used to match images in our Washington D.C. task.

ARCHITECTURES FOR IMAGE PROCESSING

SPARC, the high speed processor being jointly designed by Control Data and CMU, is in the final stages of hardware design and gate level simulation. Some hardware changes have been made since the last vision workshop, including an expansion of the crossbar switch and the addition of fast register file functional units. Software design teams at each location are beginning a cooperative effort to specify and implement a new assembler and simulator for the machine. In addition they will work on a vision algorithm package, written in SPARC assembly language, for general purpose image understanding tasks.

Researchers at CMU have already designed several prototype NMOS LSI circuits utilizing graphics software running under the UNIX operating system. Work is underway to complete a design laboratory which will allow top down design of VLSI circuits, as well as provide post-fabrication packaging and testing facilities. The laboratory is intended to allow computer scientists with a minimal understanding of solid-state physics and IC design to rapidly produce working circuits. A number of special purpose chips are expected to be designed to implement common image understanding algorithms, such as edge detectors and smoothing operators.

We recently began collaboration with Texas Instruments to jointly design and develop an all-digital programmable VLSI chip set for several low level vision operations. The paper by Eversole et. al. in this volume describes the design concepts for one of the proposed chip sets.

CONCLUSION

While the primary emphasis continues to be in effective use of knowledge in the image interpretation process, the research at CMU is tempered by the realization that we must also pay adequate attention to other relevant aspects such as computer architecture, software design, image databases, performance analysis and perceptual psychology. We continue to have modest efforts in each of these areas.

REFERENCES

Kanade, T. (1978). "A Theory of Origami World" CMU Technical Report, Department of Computer Science. September, 1978.

Kender, J. (1978). "Shape from Texture: A Brief Overview and a New Aggregation Transform", in this volume.

McKeown, D. M. and Reddy, D. R. (1977). "A Hierarchical Symbolic Representation for an Image Database" Proceeding of *IEEE Workshop on Picture Data Description and Management*, April, 1977.

Ohlander, R. (1975). "Analysis of Natural Images," Ph.D Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA. Rubin, S. (1978). "The ARGOS Image Understanding System", in this volume.

Price, K. (1976). "Change Detection and Analysis of Multi-Spectral Images," Ph.D Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.

Shafer, S. and Kanade, T. (1978 in preparation). "KIWI: A Flexible System for Region Segmentation", CMU Technical Report, Department of Computer Science.

SESSION II

TECHNIQUES I

SYNTACTIC ERROR-CORRECTING RECOGNITION OF PATTERNS AND ITS APPLICATION TO TEXTURE DISCRIMINATION

W. H. Tsai and K. S. Fu
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

Various types of pattern deformations are investigated from syntactic point of view and categorized into two major types: local deformation and structural deformation. Every observed pattern can be regarded as transformed from a pure (error-free) pattern through these two types of deformation. A local deformation is further decomposed into two steps: a syntactic deformation followed by a semantic one, the former being induced on the primitive structures and the latter on the primitive attributes. According to this deformational model, an error-correcting parsing scheme optimum in the Bayes sense for local deformations is proposed, which can utilize continuous numerical information contained in the pattern primitives. A Bayes recognition rule for pattern classification is also described. These techniques are then applied to texture discrimination, and the results show that numerical attributes contained in the primitives indeed can be fully utilized for discrimination during syntactic parsing.

1. Introduction

To recognize noisy or deformed patterns using the syntactic pattern recognition approach, error-correcting parsing and classification techniques using various decision criteria have been proposed [1-5]. Errors induced on the primitives of noisy or deformed patterns represented by strings usually are classified into three types: substitutions, deletions, and insertions. If only substitution errors are considered, the error-correcting parser is said to be structure-preserved. After an input pattern is parsed with respect to a certain pattern grammar, a quantitative measure, either deterministic or probabilistic, is used by the parser to indicate a measure of possibility that the input pattern is generated by the grammar. The decision criterion is then used to classify the input pattern as belonging to the pattern class with an extreme quantitative measure, either minimum or maximum, depending on how the measure is defined. Two most widely used decision criteria are the minimum-distance and the maximum-likelihood criteria.

Influenced by the linguistic types of representation which only adopts symbolic notations as terminals, most of the existing error-correcting parsing methods [1-5] use discrete symbols to represent structural pattern primitives. However, it happens quite often that a primitive also contains continuous semantic or numerical attributes useful for pattern discrimination purpose [18]. For such cases, obviously, these parsing methods may not be sufficient because they can not utilize continuous semantic or numerical information. To take care of both structural and numerical information simultaneously, a deformational model for pattern primitives is introduced in this paper. Based on this model, error-correcting parsing and classification techniques using the Bayes decision rule are proposed. A special decision criterion using square-error distances is derived. The least-square-error distance criterion is then used in texture discrimination where textures are characterized by tree grammars and recognized by structure-preserved error-correcting parsers (SPECP) [4].

2. Primitives with Attributes

An observed image usually can be considered as deformed from a pure image. When similar pure images are clustered as a pure pattern class, there corresponds a set of observed images each of which we will call an observed pattern. In some simple cases, the deformation, such as noise, existing in observed patterns can be eliminated by a preprocessing such as thresholding. But in general, it can not be eliminated entirely. This is why error-correcting parsings are necessary. Before a class of patterns can be described by a pattern grammar, each pattern is decomposed into some basic components called primitives. We call the description of a pattern using some fixed primitives according to some fixed pattern structure as a structural representation. A detailed study of various kinds of primitives used for pattern descriptions [8,12,18] reveals that each primitive may contain two kinds of information, namely, the syntactic information and the semantic information. The syntactic information gives a structural description of the primitive, and the semantic information provides the meaning or numerical description of the primitive. Thus, a formal description of a primitive a, either pure or observed, can be considered as a 2-tuple

$$a = (s,x)$$

where s is a syntactic symbol denoting the primitive structure of a, and $x = (x_1,x_2,\dots,x_m)$ is an m-dimensional semantic vector with each x_i $(i = 1,2,\dots,m)$ denoting a numerical or a logical attribute.

3. A Pattern Deformational Model

From previous discussions, it is clear that a pattern or its structural representation ω can be fully characterized by a 2-tuple $\omega = (S,A)$ where $A = \{a_i | i = 1,2,\ldots,n\}$ is a set of primitives used in ω and S denotes the pattern structure of ω together with implicitly assumed relations among the primitives. For discussion convenience in the following sections, we assume that the subscripts for a_i are numbered according to some fixed order which is determined by the pattern structure S; when S is fixed, then this ordering is also fixed.

Given the structural representation $\omega = (S,A)$ of a certain *pure* pattern with pattern structure S and primitive set

$A = \{a_i | a_i = (s_i,x_i),\ x_i = (x_{i1},x_{i2},\ldots,x_{iN_i}),$
$N_i \geq 0,\ i=1,2,\ldots,n\}$,

the structural representation of its corresponding *observed* pattern $\omega' = (S',A')$, with pattern structure S' and primitive set

$A'=\{a'_i | a'_i = (s'_i,x'_i),\ x'_i=(x'_{i1},x'_{i2},\ldots,x'_{iN'_i}),$
$N'_i \geq 0,\ i=1,2,\ldots,n\}$,

can be considered as being transformed from ω through a series of deformations. Our deformational model categorizes all possible deformations into two major types: *structural deformations* and *local deformations*.

I. Local deformations --- If S = S', but for some i, i = 1,2,...,n, $a_i \neq a'_i$, then we say ω' is *deformed locally* from ω. A local deformation is also called a *structure-preserved deformation*. With respect to strings, this simply means a *length-preserved deformation*.

II. Structural deformations --- if $S \neq S'$, then we say that ω' is *deformed structurally* from ω. Various types of structural deformations, such as insertions, deletions, transpositions, and permutations [2,5] have been defined according to various kinds of structural difference between S and S'.

In this paper, we deal only with local deformations, leaving structural deformations for further investigations. Let $a_i = (s_i,x_i)$ be the pure primitive deformed where $x_i = (x_{i1},x_{i2},\ldots,x_{iN_i})$, and $c_i = (t_i,z_i)$ be one of its observed versions, where $z_i = (z_{i1},z_{i2},\ldots,z_{iN'_i})$. At least two types of local deformations can be identified as following:

I. Syntactic local deformation --- This is the case when $t_i \neq s_i$. In another word, when the primitive structure is changed to another one, a *syntactic local deformation* is induced, which usually is called a substitution error.

II. Semantic local deformation --- When the local deformation on a_i does not change the primitive structure but only corrupts the semantic information, i.e. when $t_i = s_i$ but $z_i \neq x_i$, then it is called a *semantic local deformation*.

In general, we can consider a local deformation as a two-step transformation from $a_i = (s_i,x_i)$ to $c_i = (t_i,z_i)$ by the following way:

$$\begin{tikzcd}
(s_i,x_i) \arrow[r, "p(t_i|s_i)", "\text{synt.loc.def.}"'] & (t_i,y_i) \arrow[r, "q(z_i|t_i,s_i)", "\text{sem.loc.def.}"'] & (t_i,z_i) \\
\text{pure prim.}a_i \arrow[u] & \text{semi-pure prim.}b_i \arrow[u] & \text{observed prim.}c_i \arrow[u]
\end{tikzcd}$$

where $b_i = (t_i,y_i)$, called a *semi-pure primitive*, is created to denote one of the syntactically local-deformed versions of (s_i,x_i) with y_i being a *representative semantic vector* for t_i, which is created for explanatory convenience, $p(t_i|s_i)$ is the probability for $a_i = (s_i,x_i)$ to be deformed into $b_i = (t_i,y_i)$, and $q(z_i|t_i,s_i)$ is the probability or density for $b_i = (t_i,y_i)$ to be deformed into $c_i = (t_i,z_i)$. So the total probability or density for a_i to be deformed into c_i is

$$r(c_i|a_i) = p(t_i|s_i)q(z_i|t_i,s_i)$$

And given a pure pattern $\omega = (S,A)$ with $A = \{a_i | a_i = (s_i,x_i),\ i = 1,2,\ldots,n\}$, the probability or density that ω is deformed locally into a structure-preserved observed pattern $\omega' = (S,C)$ with

$$C = \{c_i | c_i = (t_i,z_i),\ a_i \xrightarrow[\text{loc.def.}]{} c_i,\ i = 1,2,\ldots,n\}$$

is

$$P(\omega'|\omega) = \prod_{i=1}^{n} r(c_i|a_i)$$

$$= \prod_{i=1}^{n} p(t_i|s_i)q(z_i|t_i,s_i)\ ,$$

if each a_i is deformed *independently* into c_i, $i = 1,2,\ldots,n$. Such independence assumption for local deformations of primitives was also considered by Grenander [14], Kovalevsky [15], and Fung and Fu [3].

4. Bayes Structure-Preserved Error-Correcting Parsers and Least-Square-Error Distance Criterion.

Given a pattern class consisting of various pure patterns which can be generated by a pattern grammar, we can, from statistical point of view, consider each pure pattern *together with all its* possible locally deformed versions as a distinct subclass of the given pattern class. Then the SPECP to be derived, which we will call Bayes SPECP, are optimum in the sense that they are, in addition to possessing syntactic parsing capability, just Bayes subclass classifiers which assign each given observed pattern, according to Bayes decision rule, to a subclass whose pure pattern has a maximum probability to be deformed into the given observed pattern.

Given an observed pattern $\omega = (S,A)$ with $A = \{a_i | a_i = (s_i,x_i),\ x_i = (x_{i1},x_{i2},\ldots,x_{iL_i}),\ i = 1,2,\ldots,n\}$ of a certain pure pattern class C *which consists of M pure patterns, each pattern* $\omega_j = (S,B_j)$ with $B_j = \{b_i^j | b_i^j = (t_i^j,y_i^j),\ y_i^j = (y_{i1}^j,\ y_{i2}^j,\ldots,y_{iM_i^j}^j)\ i = 1,2,\ldots,n\}$, we will assign ω to one of the M pure pattern subclass ω_k according to the Bayes decision rule. It is proved in [7] that this is equivalent to assign ω to ω_k *if* k is such that

$$-\ell n\ \lambda_k = \min_{j = 1,2,\ldots,M} (-\ \ell n\ \lambda_j),$$

where

$$-\ell n\ \lambda_j = -\sum_{i=1}^{n} [\ell n\ p(s_i|t_i^j) + \ell n\ q(x_i|s_i,t_i^j) - \ell n\ P(\omega_j).$$

We call the term $-\ell n\ \lambda_j$ the Bayes distance $B(\omega,\omega_j)$ *from* ω *to* ω_j*, and the term* $-\ell n\ \lambda_k$ the minimum Bayes distance $B(\omega,C)$ from ω to pure pattern class C.

With the Bayes distance as defined above, the Bayes structure-preserved error-correcting parser constructed from the pattern grammar G_c for a given pure pattern class C, is used to search *for a given input observed pattern* ω *a* pure pattern ω_k accepted by G_c with a minimum Bayes distance $B(\omega,\omega_k) = B(\omega,C)$ during the error-correcting parsing. Since the parsing is performed on each primitive at least once, there is no problem in computing the term $\sum_{i=1}^{n} [p(s_i|t_i^j) + \ell n\ q(x_i|s_i,t_i^j)]$ in $-\ell n\ \lambda_j$ during the parsing procedure. But getting the a priori probability $P(\omega_j)$ for the pure pattern ω_j during the parsing procedure is on the contrary not so obvious. The key is to use a stochastic grammar for pattern class C, because a stochastic grammar can be used to generate pattern occurrence probabilities during parsing [8]. Using stochastic grammars and the minimum-Bayes-distance criterion, two Bayes SPECP's, one for string languages and the other for tree languages, have been proposed by Tsai and Fu [13].

Finally, we propose in the following a new criterion, namely, the least-square-error (LSE) distance criterion for the SPECP, which is a special case of the minimum-Bayes-distance criterion but is useful for semantic local deformations.

If we can assume that the observed semantic vector in a primitive is normally distributed, and no syntactic local deformation occurs, then it is possible to derive the Bayes distance between *a pure pattern* $\omega = (S,B)$ and one of its normally deformed observed patterns, $\omega' = (S,A)$. Let $A = \{a_i|a_i = (s_i,x_i),\ x_i = (x_{i1},x_{i2},\ldots,x_{iN}),\ i = 1,2,\ldots,n\}$ and $B = \{b_i|b_i = (s_i,w_i),\ w_i = (w_{i1},w_{i2},\ldots,w_{iN}),\ i = 1,2,\ldots,n\}$, and assume that component random variables x_{ij} of x_i are all independently and normally distributed with mean w_{ij}, and variance σ_{ij}^2 $j = 1,2,\ldots,N$, (An example for this case happens when every x_{ij} is corrupted with random noise with zero mean and variance σ_{ij}^2) and that the pure pattern ω has the same probability to occur as any other, so that $P(\omega_j)$ is a constant for every pure pattern ω_j. Then Bayes distance from ω' to ω can be easily derived as

$$B(\omega',\omega) = K + \sum_{i=1}^{n} \sum_{j=1}^{N} [\frac{1}{2} (\frac{x_{ij} - w_{ij}}{\sigma_{ij}})^2 + \ell n\ \sigma_{ij}]\ ,$$

where K is a constant. As far as discrimination is concerned, we can define the normalized square-error distance as

$$B_1(\omega',\omega) = \sum_{i=1}^{n} \sum_{j=1}^{N} [(\frac{x_{ij} - w_{ij}}{\sigma_{ij}})^2 + 2\ell n\ \sigma_{ij}]\ ,$$

and the (unnormalized) square-error distance as

$$B_2(\omega',\omega) = \sum_{i=1}^{n} \sum_{j=1}^{N} (x_{ij} - w_{ij})^2$$

which is valid under a further assumption that all $\sigma_{ij} = 1$. A SPECP using the normalized or unnormalized least-square-error (LSE) distance criterion is called a normalized or unnormalized LSE SPECP. They will be used later in texture discrimination.

5. Bayes Error-Correcting Recognition System

Given m pattern classes $C_1,C_2,\ldots,C_m$ of pure patterns and their pattern grammars $G_1,G_2,\ldots,G_m$, after a given input observed pattern ω is parsed by all the Bayes SPECP of the grammars, we get a set of minimum Bayes distances $B(\omega,C_1)$, $B(\omega,C_2)$, $\ldots$, $B(\omega,C_m)$. Actually, these distances are just the negative logarithms of the conditional probabilities or densities of ω given that $\omega \varepsilon C_i$, or

$$p(\omega|C_i) = \mathrm{EXP}[\ -\ B(\omega,C_i)]\ ,$$

$i = 1,2,\ldots,m$. Our classification problem is to assign ω to one of these m classes, which has a highest possibility to accept ω as its observed pattern.

Again, we can apply the Bayes decision rule to get

$$P(C_\ell|\omega) = \max_{i=1,2,\ldots,m} P(C_i|\omega) \text{ decide } \omega \rightarrow C_\ell,$$

or

$$P(\omega|C_\ell)P(C_\ell) = \max_{i=1,2,\ldots,m} p(\omega|C_i)P(C_i) \text{ decide}$$

$$\omega \rightarrow C_\ell$$

where $P(C_i)$ is the a priori probability for pattern class C_i, $i=1,2,\ldots,m$. We call this interclass Bayes classifier together with the interclass Bayes SPECP a Bayes error-correcting recognition system, compared to the maximum-likelihood classification system set up originally by Fung and Fu [3]. Such a Bayes error-correcting recognition system essentially has also been proposed by Lu and Fu [5] and Fung and Fu [17], but, as mentioned in the Introduction, the error-correcting capability for substitution errors of their system can only take care of syntactic local deformations. The proposed system here can also handle semantic local deformations.

6. Application to Texture Discrimination

The world is rich in texture scenes, and texture analysis and discrimination are important in image understanding. While most researches concentrated on statistical approaches in the past years [9,10,16], recently a syntactic approach has been successfully applied to texture analysis by Lu and Fu [11]. In their approach, texture patterns are thresholded and divided into windows with some predetermined size. Each pixel or a small cell of pixel array together with its average gray value is chosen as a primitive, and each window is transformed into a tree representation according to a tree structure which can be arbitrarily chosen but is fixed through the later processing. Tree representations of a given texture are used to infer a tree grammar which, when made stochastic, can describe noisy or distorted texture patterns. Finally, two kinds of error-correcting tree automata using the minimum-distance and the maximum-likelihood criteria [4] are adopted as the tree parsing scheme for texture recognition.

When such a syntactic approach is applied to real world texture discrimination, several problems arise which are worth further investigations. For example, identical textures but in different orientations usually need different tree grammars to characterize. In addition, if too large window sizes are used, the parsing efficiency of each window and segmentation accuracy at texture boundaries will be decreased. Furthermore, gray level difference is a good discriminant factor for texture recognition and it is desirable to include into the recognition scheme all gray levels without thresholding to improve discrimination results. But if too many gray levels are used in the pixel primitives, the resulting tree grammar could become very complicated.

To solve the first problem, i.e., to avoid constructing a complicated grammar to cover all possible texture orientations, Tsai and Fu [6] propose the use of a direction detection technique and transformational grammars to reduce the number of orientations which should be covered by the texture grammar. They also propose an algorithm to solve the window size selection problem. The algorithm can choose for a given set of texture pictures a common square window size such that the resulting texture grammars corresponding to this size will have high discriminating capabilities for texture recognition. For details see [6]. Our main concern here is the third problem, i.e., can we utilize continuous grey level information existing in the textures as a discriminating factor during the syntactic analysis of texture structures by tree grammars?

The solution to this problem is to use the least-square-error distance criterion for error-correcting parsings of texture windows. More specifically, we consider the gray value associated with each pixel primitive as a semantic random variable and the whole texture picture as a random field [19]. Since every primitive is of the same structure -- a pixel, the deformations induced on the pixel primitives are all semantic ones. And since the textures we process (agricultural area image) is well-structured, they are assumed to be corrupted by normally distributed random noise. Therefore, we can use the LSE SPECP as the intraclass recognizer.

7. Experimental Results on Segmentation and Recognition of Agricultural Area Pictures

Our experiment on the segmentation and recognition of agricultural area pictures is divided into two parts: the training stage and the discrimination stage. The data used is shown in Fig. 1, which consists of 4 kinds of texture patterns: cotton in the upper left, mangos in the lower left, wax apples in the upper right, and papayas in the lower right, which are denoted as C, M, W, P, respectively. Our purpose is to segment this picture into 4 different texture areas. Since each texture belongs to a unique kind of plant, such segmentation itself is also a recognition procedure.

In the training stage, with Fig. 2 as the input, the window size selection algorithm is used to infer a window size of 9x9. Four representative grammars are also inferred with mean gray values of the plants and backgrounds as primitive attributes.

In the discrimination stage,windows in Fig. 1 are rotated after direction detection, and classified by using the unnormalized LSE SPECP's of the grammars. The result is shown in Fig. 3

in which each window is generated by the texture grammar with the least-square-error distance. Totally 7 windows are misclassified with recognition accuracy at about 89.1%. Normalized LSE SPECP's are used next to analyze Fig. 1. The result is shown in Fig. 4 in which the number of misclassified windows is, as expected, reduced to 4, with a 93.8% recognition accuracy.

BIBLIOGRAPHY

1. Aho, A.V. and Peterson, T.G., "A Minimum Distance Error-Correcting Parser for Context-Free Languages," SIAM J. Comput., Vol. 1, pp. 305-312, Dec. 1972.

2. Thompson, R.A., "Language Correction Using Probabilistic Grammars," IEEE Trans. on Comput., Vol. C-25, No. 3, Mar. 1976.

3. Fung, L.W. and Fu, K.S., "Stochastic Syntactic Decoding for Pattern Classification," IEEE Trans. on Comput., Vol. C-24, No. 6, July 1975.

4. Lu, S.Y. and Fu, K.S., "Structure-Preserved Error-Correcting Tree Automata for Syntactic Pattern Recognition," IEEE Conf. on Decision and Control, Dec. 1-3, Clearwater Beach, FL, 1976.

5. Lu, S.Y. and Fu, K.S., "Stochastic Error-Correcting Syntax Analysis for Recognition of Noisy Patterns," IEEE Trans. on Computers, Vol. C-26, No. 12, Dec. 1977.

6. Tsai, W.H. and Fu, K.S., "Image Segmentation and Recognition by Texture Discrimination: A Syntactic Approach," Proceedings of the 4th International Joint Conference on Pattern Recog., KYOTO, JAPAN, Nov. 7-10, 1978.

7. Tsai, W.H. and Fu, K.S., "A Pattern Deformational Model and Bayes Error-Correcting Recognition System," Proceedings of the International Conference on Cybernetics and Society, Tokyo, Japan, Nov. 3-7, 1978.

8. Fu, K.S., Syntactic Methods in Pattern Recognition, New York Academic Press, 1974.

9. Haralick, R.M., Shanmugam, K., and Dinstein, I., "Texture Features for Image Classification," IEEE Trans. on SMC, Vol. SMC-3, No. 6, Nov. 1973.

10. Weszka, J.S., Dyer, C.R., and Rosenfeld, A., "A Comparative Study of Texture Measures for Terrain Classification," IEEE Trans. on SMC, Vol. SMC-6, No. 4, April 1976.

11. Lu, S.Y. and Fu, K.S., "A Syntactic Approach to Texture Analysis," Compt. Graphics and Image Processing, June 1978.

12. Shaw, A. C., "A Formal Picture Description Scheme as a Basis for Picture Processing Systems," Inform. and Control, Vol. 14, 9-52 (1969).

13. Tsai, W.H. and Fu, K.S., "A Pattern Deformational Model and Bayes Error-Correcting Recognition System," Purdue University, TR-EE 78-26, May 1978.

14. Grenander, U., "A Unified Approach to Pattern Analysis," in Advances in Computers, Vol. 10, New York: Academic, 1970.

15. Kovalevsky, V.A., "Sequential Optimization in Pattern Recognition and Pattern Description," in Proc. Int. Fed. Info. Process. Congr., Amsterdam, the Netherlands, 1968.

16. McCormick, B.H. and Jayaramamurthy, S. N., "A Decision Theory Method for the Analysis of Texture," Int. J. of Compt. and Inf. Sci., Vol. 4, No. 1, 1975.

17. Fung, L.W. and Fu, K.S., "Syntactic Decoding for Computer Communication and Pattern Recognition," Purdue University, TR-EE 74-47, Dec. 1974.

18. You, K.C. and Fu, K.S., "Syntactic Shape Recognition Using Attributed Grammars," 8th Annual Automatic Imagery Pattern Recognition, April 3-4, 1978, Gaithersburg, Maryland.

19. Rosenfeld, A. and Kak, A. C., Digital Picture Processing, New York: Academic Press, 1975, Sec. 2.4.

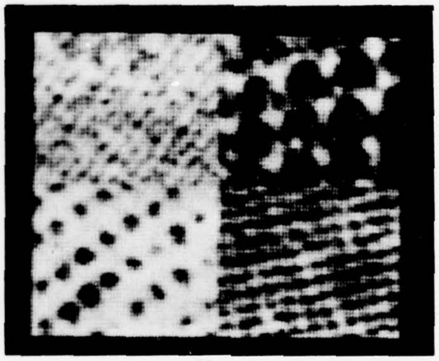


Fig. 1

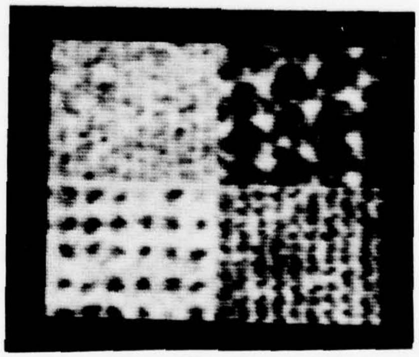


Fig. 2

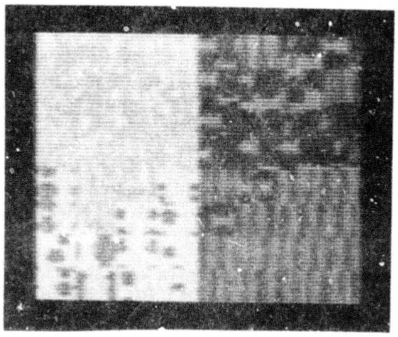


Fig. 3

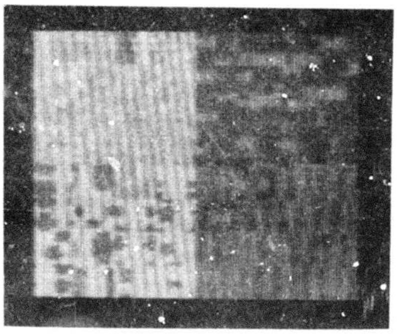


Fig. 4

ADVANCES IN SHAPE DESCRIPTION WITH APPLICATION TO THREE-DIMENSIONAL AIRCRAFT RECOGNITION

T. Wallace, P. A. Wintz, and O. R. Mitchell

Purdue University
West Lafayette, Indiana

ABSTRACT

Fourier descriptors (FDs) are well known global shape descriptors. Previous shape recognition algorithms based on FDs have suffered from excessive computation, or loss of shape information. In this paper, we discuss a more efficient algorithm based on better understanding of the relationship between a shape and its FD. Shape representation is considered as part of the complete algorithm, and a new definition of chain code error is presented which has the property of invariance to sampling grid resolution. The importance of this error to FD computation is shown, and the results of experiments in three-dimensional aircraft recognition are presented.

1 INTRODUCTION

In a recent workshop paper [11], theoretical results were presented regarding an algorithm for recognizing three-dimensional objects using Fourier descriptor (FD) features derived from their boundaries. Preliminary experimental results were described at that time which indicated that the approach was feasible. Since then, extensive experiments have been conducted using the algorithm described in [11], as well as several modifications of that procedure. Some additional theoretical considerations are discussed in this report, which also presents the best results achieved to date. The experimental effects of varying some the parameters associated with the algorithm are shown in tabular form.

As discussed in [11], our present algorithms are based on the original FD definition of Granlund [1], who defined the complex Fourier series which is the basis of the method. Recall that the FD of a contour is computed by tracing the contour in the complex plane, and then expanding the resulting function in a Fourier series. The function is assumed to be periodic, i. e. the contour can be traced repeatedly. The actual features used by Granlund were certain non-linear functions of the original Fourier series coefficients. It was not clear how the shape information transformed from the original Fourier series to these "Fourier descriptors." An improvement was made by Persoon and Fu [2] [3], and by Richard and Hemami [5], in that the actual Fourier series coefficients were used as features, so that the information was contained in an easily understood Fourier vector. The main problem was that distance computations now required searching for an optimum starting point, orientation, and size which mimimized the distance between Fourier vectors. The computation to do this was excessive if there were many classes, such as in the three-dimensional case. The fastest method [5] required two FFTs for each comparison of an unknown shape to a library shape. The result was the definition of "suboptimum" comparison techniques which generally performed reasonably well in the experiments reported, but again tended to obscure the actual shape information being used for classification.

Our technique is based on a better understanding of the relationship between a shape and its FD representation. We define a standard orientation, starting point, and size by a normalization procedure performed entirely in the frequency domain. After this, distance computations can be made by simple comparisons between normalized feature vectors. In addition, a big advantage to these feature vectors is that they have the property of linearity. While common in mathematics, linearity is rarely observed in features used for shape description. Those techniques which initially may have potential for linearity generally have some non-linear function defining the actual feature vector [1]-[5]. We exploit the linearity property in a scheme which uses linear interpolation to define an actual continuum of projections in three space, although there are only a finite number of samples in our projection library.

The problems with past Fourier descriptor algorithms have obscured the problem of representing a contour taken from a sampled image. A chain code is generally taken to represent the contour acceptably, but it is shown here that a chain code representation error defined as a difference between the chain code length and the actual length of part of a contour presents significant problems not generally considered by researchers working with shape descriptors. A certain lack of generality in many shape recognition experiments is shown to reduce the effect of chain code representation error. Techniques are presented for reduction of this error, and the results of more general experiments are presented.

2 CHAIN CODE REPRESENTATION ERROR

2.0 Definition

One source of error in computing Fourier descriptors from sampled image contours is chain code representation error. The contours are defined on the sampling grid of the image, and will generally consist of either a four-neighbor chain code, or an eight-neighbor chain code. The four-neighbor code represents the contour by vertical and horizontal line segments only, while the eight-neighbor includes the diagonals. Approximating a continuous contour by a piecewise linear function such as a chain code presents non-trivial theoretical problems. The first thing to observe is that these procedures do not result in a uniform sampling of the contour, since those portions of the actual contour which are not directly representable by a chain code are shorter than the perimeter of the corresponding chain code. This can be viewed as an error in sampling density, and can result in significant variation in performance as a function of orientation in the picture. Features derived from a right triangle, for example, in which the legs fall right on the chain code grid and the hypotenuse does not, may differ significantly from features derived from the same triangle oriented in the picture such that the hypotenuse falls right on the grid, and the legs do not. (Fig. 1)

Define a line segment in the sampling grid to be a straight line connecting any two points in the grid. Consider the difference of the length of the best chain code representation of a line segment and the actual length of that line segment. Take the ratio of that difference to the actual line segment length. We will define the "chain code representation error" for a chain code as the maximum of such ratios taken over all possible line segments. This number also describes the sampling error over the line segment. Note that the restriction that the line segment must connect two points in the grid still allows the angles to range over all the rational numbers since the sampling grid is infinite.

The error is easily computed for the commonly used four- and eight-neighbor chain codes, and proves to be $\sqrt{2} - 1$ and $\sqrt{1/5}+\sqrt{2/5} - 1$ respectively. (The simple diagonal is the worst case for the four-neighbor code, and the diagonal of a two unit by one unit section of grid is the worst case for the eight neighbor code.) The four-neighbor code thus exhibits more than five times the error of the eight-neighbor code.

2.1 Effects of Chain Code Representation Error

Although it might seem that the effect of this error would be difficult to determine theoretically, it so happens that we have proved a theorem useful in that connection. The theorem applies to the situation in which two Fourier descriptors A and B have been computed from two contours sampled at n points. There is no requirement that the sampling be uniform. The conclusion is that the mean square distance obtained by squaring the real and imaginary parts of the difference coefficients (A-B) and summing over n, is proportional to the space domain distance obtained by taking the square of the geometric distance between each of the n original sample locations, again summing over n. (The proof is a straightforward application of Parseval's theorem.) There is a clear order to the coefficients in the frequency domain since each represents a different frequency, and it should be noted that there is also a corresponding ordering in the space domain in which the first point is simply the point which appears first in the inverse FFT vector.

It is complicated to make quantitative statements about the error resulting from two shapes exhibiting this sampling error, but it is fairly easy to compute the error between one uniformly sampled contour and one example of worst-case sampling error. It is clear that the actual worst case behavior would be roughly twice as bad as that obtained by this analysis, since the actual worst-case situation involves a library contour originally oriented to create error opposite to that in the unknown contour. The slightly subtle point here is that though the original chain codes were oriented differently, the normalization procedure orients them similarly, so that an inverse transform of the normalized FDs (NFDs) would look about the same, except for this sampling error.

The normalization procedure generally chooses a vertex as the starting point, so assume that we are comparing the two triangles of Fig. 1, and that the starting point is the left hand vertex. The sampling density error is zero for the first of the n samples. The error for sample 2 is .08, where we arbitrarily take the unit distance to be the length of one chain code non-diagonal. When this error is squared, we get .0064, which looks insignificant enough. The problem is that this error is cumulative, so that the error for the third point is .16, which when squared is .0256. The 21st point gives us an error of 2.56, and it is clear that classification accuracy is in jeopardy. The interpretation of this situation is simply that when the two triangles are compared in this point by point fashion, the triangle exhibiting more dense sampling on the first segment traversed has its points falling behind those of the other triangle. Although the triangles may be registered fairly well, the point by point distances reflect the sampling error more than the shape similarity and registration. Note that this error is independent of sampling grid size, so that increasing the sampling resolution will not reduce this error.

Recall that the distance measures used in the classification process were a mean square distance and an absolute value distance. The performance was slightly better with the absolute value distance, but the differences were minor. It certainly appears unlikely that any significant improvement in classification accuracy using the m. s. measure would not parallel a corresponding improvement in absolute value performance. The experimental results of part 4 substantiate this.

The expected value of the error would not be nearly as large as this worst case behavior, since the worst case occurs when there is a long edge or edge sequence which exhibits the maximum sampling error, followed by a long sequence which exhibits zero sampling error (falls right on the chain code grid). If the actual contours under analysis generally consisted of smaller segments which alternated sampling error, the error would not accumulate for too many points before it started to decrease on a segment exhibiting error of the opposite type.

2.2 Representation Error Reduction

Experimental results presented later in this paper indicate that despite chain code representation error, good classification accuracy can be obtained. However, a method has been developed to greatly reduce the problem, and results using this method indicate a small but not insignificant improvement in classification accuracy. Perhaps more important than the improvement in the algorithm's performance on the present data is the protection afforded against possible future appearances of "worst case" data.

Recall that the Fourier descriptor of a chain code contour is computed by first converting the chain code coordinates to x-y coordinates in the complex plane, choosing an arbitrary starting point. Next the perimeter of the contour is computed and the contour is uniformly resampled to obtain a sample vector of length a power of two. The FFT is used to compute the Fourier descriptor, and the normalization procedure is performed.

The important thing to note is that later classifications use no more than 30 frequencies of the FFT vector. In fact, to speed up the normalization procedure, the FFT vector is truncated before normalization. The frequencies used in this classification will not be significantly affected if a convolution with a small window is performed on the data before it is resampled uniformly. We can view the original x-y coordinate representation derived directly from the chain code as the sum of the actual contour and a noise sequence. As observed above, the major problem is that later classifications can suffer from accumulations of this noise sequence, although the noise sequence itself is not of great amplitude. The solution is to apply a non-recursive averaging type digital filter to this sequence, greatly reducing the noise. Note that we are filtering a complex sequence, so we are really filtering two sequences, one real (x), and one imaginary (y). The noise is such that originally the slope of the line connecting two adjacent points is constrained to be a multiple of $\pi/4$. After filtering, there is no such restriction.

The effect on the actual contour is not easy to describe exactly, since it amounts to a convolution with a window of varying size due to the $\sqrt{2}$ ratio between a diagonal chain code link and a non-diagonal. We can say, however, that a small enough window will not have very deleterious effects, regardless of its slight space-variant character. Intuitively, looking back at Fig. 1, the effect on the actual contour sequence (the triangle) should be nil in the middle of each edge, with a slight rounding expected at each vertex. However, the noise sequence can be expected to decrease greatly with a window as small as several points wide.

Figs. 2-11 show the effects of filtering on a representative contour. Fig. 2 is the original chain code representation, and Figs. 3-11 show the effects of filtering with various window sizes and shapes. Fig. 12 is a chain code representation of the same contour as Fig. 2, with a slight additional rotation. The two chain code representations exhibit representation error on different edges. Fig. 13 shows the effect of filtering on the chain code of Fig. 12. Fig. 13 is much more like Fig. 4 than Fig. 12 is like Fig. 2, illustrating the advantages of chain code representation error reduction. Table 1 shows algorithm performance both with and without various amounts of digital filtering.

3 ANOTHER LOOK AT NORMALIZATION

The normalization procedure discussed in [11] has been proven effective in recent experiments. The basic idea has been to define a standard size, orientation, and starting point for any contour by working in the frequency domain. The size has been easily normalized using the magnitude of the fundamental frequency coefficient, and the orientation and starting point normalizations have been performed simultaneously in order to achieve zero phases for the two coefficients of largest magnitude. Choosing coefficients of large magnitude has proven effective in combating noise.

The only problem which arises in this method occurs when the coefficient of second largest magnitude is not the second harmonic. (The fundamental always has the largest magnitude.) It has been shown [11] that if the coefficient of frequency k is used for normalization, there exist |k-1| distinct orientation/starting point combinations which satisfy the requirement that A(1) and A(k) have zero phase. Our previous approach to this problem [11] makes use of a third coefficient to resolve this ambiguity. The third coefficient is chosen to be as large as possible, but there are several restrictions on which frequencies may be used to resolve the ambiguity associated with normalization by A(k).

It is not clear that the same coefficient will be used to resolve this ambiguity for both the library FD and the similar unknown FD, so a potential normalization problem exists. One solution is to retain the coefficients used to normalize each library NFD, and to normalize the unknown data several different ways so that comparisons can always be made between similarly normalized FDs. In practice, this might require five to fifteen different normalizations of the unknown FD, although the algorithm might not be slowed down significantly, since distance computations consume most of the processing time. A more elegant solution is to examine the |k-1| possible normalizations more closely, employing a more sophisticated criterion for choosing one. Such a method has been developed, and involves normalizing the vector each of the |k-1| possible ways and then optimizing some function of each possible NFD. The best results to date have been achieved by maximizing the function

$$\sum_{i=1}^{N-1} \mathrm{Re}[a(i)]\,|\mathrm{Re}[a(i)]| \qquad (1)$$

Table 2 shows the effects of various normalization schemes on classification accuracy in the experiment described below.

4 EXPERIMENTAL RESULTS

The experiments used to test our three-dimensional recognition/estimation algorithm are described in detail in [11]. The only difference between the experiments described there and the present ones concerns the density of projections used to represent each aircraft. The previous experiment used 99 projections to represent each aircraft over a hemisphere, and the more recent work reported here uses 143 projections per aircraft. Note that the projection density is still 9.9 times lower than that used by Dudani et al [4], in a similar experiment.

Briefly, a set of six aircraft (Fig. 14) was synthesized using graphics programs which also enable a projection to be obtained at any angle. A set of library contours was computed consisting of 143 projections of each of the six aircraft. Fifty unknown contours were computed for each of the six aircraft by taking projections at random angles. All of the projections used in the library were taken from above the aircraft, as were the unknown contours. One experiment was performed, however, in which the reference set remained the same, but the unknown contours were taken 50 from above and 50 from below, for each aircraft.

The actual experiment proceeded as follows. First, the NFDs of all the contours were computed. Then a linear transformation was performed on the data, based on the eigenvalues and eigenvectors of the autocorrelation matrix. The data dimensionality was thus reduced from 30 to 5. Next each unknown NFD was compared with the library of NFDs, using either a mean square or absolute value distance measure. The k nearest library NFDs were found, with the restriction that library NFDs more than d times the distance to the nearest library NFD were not considered. Typical values for k and d were 1 - 10, and 1.3 - 2.0 respectively. The nearest projections thus found were then used in an estimation procedure which looks in the sectors adjacent to each close projection, and performs linear mean square estimation as described in [11]. This effectively defines a continuum in NFD space, which interpolates between the samples represented by the original library projections. The distance to the nearest library projection or interpolated projection is minimized and the aircraft and orientation corresponding to that minimum projection are taken to be those of the unknown projection.

Tables 1-5 present the experimental results achieved. The estimation performance is noteworthy in addition to the classification performance.

5 ANALYSIS OF EXPERIMENTAL RESULTS

Tables 1-5 show the performance of this algorithm as a function of signal to quantizing noise ratio, estimation parameters, distance measure, normalization method, and digital filtering window. The unknown data identified as "Data 1" consists of unknown projections taken at random orientations, but in which the orientations relative to the sampling grid are similar to those of the nearest library projections. "Data 2" consists of the same data as "Data 1", except for an additional random rotation. This insures that the unknown projections are also at a random orientation with respect to the sampling grid. Figs. 15, 16 and 17 show unknown contours representing the various resolutions.

The maximum classification accuracy achieved for completely general data was 88.0 %. We believe that this is an excellent result for data of this type and resolution (128x128), but there is no reason to believe that this figure cannot be improved. In fact, judging from our experience with the effects of increasing library projection density, we believe that an increase in projection density of 30 to 40 % would probably push this figure well into the 90 % range. The present density was chosen since it illustrates all of the theoretical analysis presented in this paper. If a greater density had been chosen, many of the tabulated results would show classification accuracies in the 90 % range and it would be more difficult to observe the effect of varying window sizes, varying normalization procedures, etc.

It is clear that generally the estimation procedure is effective in improving classification accuracy. Also, generally, the absolute value distance measure is slightly superior to the mean square distance measure. Finally, as noted above, the digital filtering to reduce representation error and the use of normalization method 2 help classification accuracy.

One apparent anomaly in the results concerns the lowest resolution data which is adversely affected by the 4 % rectangular window digital filtering procedure. The explanation here is two-fold. First, the window is only approximately of width 4 %, since there is a minimum width of 3 points. Since many of the chain codes of this 32x32 data are only of length 50 or so, we have an effective width of 6 %, which is sub-optimum. In addition, this data is of such low resolution that resolution is to be prized above elimination of chain code representation error. Since the filter rounds some of the corners of these contours, the blurring effect is too great to be tolerable. The analysis above indicating that a "small" window would have negligible effect is still reasonable, but with this data a window of any width greater than 1 is probably not "small."

We believe that there is one approach to reducing chain code representation error which would not exhibit this drawback with very low resolution data. Instead of using a linear filtering procedure, one could employ a non-linear procedure which identifies segments of chain code which represent a straight line, and thus construct a piecewise linear approximation to the original contour. This approximation would not suffer from any vertex rounding effect at all, and might simulate the process by which a human observer would extract the original contour from a chain code representation. Similar problems have been considered before [7], [8], [9] and one of the existing algorihms can probably be used or slighty modified for this purpose. Data which is not well represented by piecewise linear models would not be able to take advantage of reduction to that form. However, such data would also not be likely to exhibit noticable amounts of chain code representation error. No such algorithm has been implemented to date, but one might be worth considering in a practical system which expected low resolution input.

Another approach to the problem of chain code representation error may be found in work with generalized chain codes [10]. These techniques result in piecewise linear contours in which the pieces are not restricted to as few as four or eight angles, and the next point is located in a square "ring" of size greater than that used by conventional chain codes. However, the complexity of such schemes is greater than the complexity of the simple digital filtering algorithm described above, so where sufficient resolution is available, digital filtering of conventional eight-neighbor chain codes is probably preferable.

Past research into chain codes has been concerned with such applications as coding and map generation, and our chain code error definition does not seem to be of more utility than conventional ones in these applications. These reseachers tend to analyze chain code errors by 1) their appeal to human observers and 2) the area error which results in coding certain silhouettes by various chain codes. The representation error defined above is more closely related to the opinions of human observers, but it is nonetheless more amenable to quantitative analysis than the area measures! This follows from the fact that the area error in coding a specific silhouette can be made arbitrarily small by using a higher resolution grid, but the chain code representation error has an approximately constant value which is independent of grid resolution. That is, given an actual contour oriented so that not all of its segments are perfectly represented by a chain code, the sampling error created by chain code representation error will remain approximately constant as the resolution of the sampling grid increases.

In comparing the accuracies achieved here with those of other researchers, it is probably safe to use the numbers associated with "Data 1" rather than "Data 2,", unless the other experiments specifically consider the problem of rotating unknown projections randomly. We believe that unknown data oriented similarly to nearest library contours is the type of data generally used in experiments such as those reported in Dudani et al [4]. Of course, it is possible that this additional rotation would not affect other features in the same way as it does our NFDs. The only way to resolve the issue would be to perform additional experiments using various features.

Given a chain code representation of the outline of an aircraft projection, the times to compute the normalized FD are about .5 sec., .9 sec., and 1.8 sec. for 32x32, 64x64, and 128x128 images respectively. The NFD is then classified and its orientation estimated in about 1.8 sec. These times are for a PDP 11/45 with floating point hardware. The program itself is written in fortran and is a research tool rather than a highly efficient implementation of the algorithm.

Most of the time spent computing the FD is associated with the FFT itself. The obvious way to speed this up is by the use of array processing hardware. Most of the time spent classifying the FD is associated with computing distances to the unknown NFDs. There should be few problems involved in partitioning the library set of FDs into ten or so overlapping classes based on the values of one or two FD coefficients. An order of magnitude classification speedup could result from comparing each unknown NFD to only those library NFDs in its class.

6 CONCLUSIONS

The Fourier descriptor has been a popular method of shape description in recent years, but an efficient method of extracting all of the shape information has been lacking. The performance of this algorithm without the estimation procedure shows that our normalized FD is a highly effective feature for shape description. In addition, the unique interpolation properties of Fourier descriptors enable a much higher level of estimation performance than competing methods.

Since shape description algorithms using contour information have been undergoing development in recent years, more attention has been paid to the shape algorithms per se than to the problem of chain code representations. Another reason why this problem can go unnoticed in the research environment is the natural tendency to compare unknown contours to library contours in which the original data is oriented in the image similarly to the library data. The additional degree of generality afforded by performing an additional rotation to the unknown data is not even always easy to achieve. Those researchers who use a model-tv camera setup to generate their data might have to rotate their camera randomly after each unknown contour is observed to achieve this! This is clearly not convenient, nor is it clear why one should bother. If the theory states that a feature is invariant to rotations, and experiments show this invariance with simple shapes, there is no obvious reason to attempt a possibly difficult experiment to verify that rotation is not a problem even in more advanced experiments. It is also likely that the effect of providing an additional rotation to unknown data, if any, will be to reduce classification accuracy. In our case, the data is generated graphically, and obtaining completely general data orientation involves no more than the addition of a couple of dozen lines of code and an insignificant amount of computation. This can certainly be viewed as another advantage of the computer graphics data generation approach.

FUTURE RESEARCH

Global features have one major problem which cannot be solved by improvements to existing algorithms. This is simply that these features are all affected by any change in the shape under analysis. If the segmentation procedure used in processing images to extract shapes for analysis fails to extract a major part of the object, there is little hope for recognition of that object using global feature methods. In satellite imagery, for example, clouds frequently cover a significant part of an object of interest. A human photointerpreter can probably still recognize the object from its partial outline, but any global feature based automatic recognition technique will fail to identify it.

It is clear that if we want automatic machine recognition of shapes to rival the performance of human observers, we must provide some method of identifying partial shapes as similar to part of a known shape. Some form of local feature must be used to accomplish this.

The main problem with using local features appears when classification methods are considered. Most previous work has used a syntactic approach to the problem, in which a grammar is derived for each pattern class, and certain rules, or "productions" are used to map the original features or "primitives" to a final classification. This procedure usually progresses through intermediate classifications of primitive combinations.

While these methods have shown promise in various applications such as those discussed by Fu [12], their biggest problem involves the lack of an effective grammatical inference algorithm. Such an algorithm would enable a machine to infer the grammar of a class of patterns automatically, based on a set of training samples. The lack of such a procedure has forced proponents of the syntactic approach to either develop the grammars themselves, or else use a man-machine interactive system to find appropriate grammars.

This does not prove a serious drawback to the method when the number of classes, and hence grammars, is relatively small. However, the three-dimensional problem often requires a description of the object which consists of hundreds of projections. Since these projections define many classes of patterns, the labor required to derive appropriate grammars becomes prohibitive. This problem is so difficult that no one has yet performed a general syntactic three-dimensional recognition experiment comparable to those which have been performed using global features [4],[5]. While slow progress is being made in this area, a breakthrough does not appear to be imminent.

Despite these problems, the local feature method appears to be the only way to effectively recognize parts of shapes in imitation of human observers. We plan to study the structures of two- and three-dimensional shapes in order to derive a local shape descriptor which can be classified with a hybrid structural/statistical technique. The existing statistical methods compare like features and look for a minimum distance or weighted distance. Existing structural (syntactic) methods reduce the features to intermediate features, and eventually to the classification itself. We plan to develop a method of computing a distance similar to those which result from present statistical algorithms, but in which the structure of the shapes being compared is exploited to facilitate computation of the distance. This structural analysis is necessary, for example, when local descriptions result in feature vectors of different dimension. Not only will this procedure enable general three-dimensional recognition experiments to be performed, but experiments even more general than those previously attempted can be performed in which partial silhouettes are classified.

There are several major problems to be solved before such an experiment can be attempted. First, it is clear that the boundary of the shape under analysis will be used to derive the local features. Use of the entire shape, or even its centroid, will tend to give problems when the partial contour recognition problem is attacked. The expansion of the boundary will require the usual two steps of segmentation and description within each segment. This problem has been investigated before, but this application requires not only a good representation of the contour, but also a representation which is amenable to some computation of distances between descriptors.

The resulting features cannot be compared on an equal basis, as can most global features, since those parts of the shape which are most important to the overall shape must be recognized as such. The major local features must be identified for use in computing a realistic distance, since the resolution of the reference descriptors may be greater than that of the unknown descriptors, It is not at all trivial to determine which local parts of a contour add up to an important part of the contour, and which parts represent minor detail which should not be weighted heavily in distance computations. If a polynomial approximation technique is adopted, for example, it is difficult to determine from the polynomial approximations to each segment the importance of that expansion to the entire contour.

Another problem concerns the segmentation of the original contour. Even if the relative importance of local features is understood, it is difficult to achieve the same segmentation for two similar contours. There are obvious problems in comparing two segments of contour which do not have like starting and ending points. If the starting and ending points are known, some procedure to compare segments along their common length can be imagined, but of course it is difficult to determine these points.

Another major problem associated with this approach is the computer programming required. Many engineers have limited experience in programming, and this problem can quickly become unwieldy at best, from a programming standpoint. Global feature classifications generally are much easier to program, and this may partially explain the lack of published work on local feature classifications using statistical distances.

Despite these problems, the promise of local feature shape description is great enough to warrant a major research effort. We believe that an algorithm of the type described above will be the first local feature algorithm capable of performing a general three-dimensional experiment. In addition, there is good reason to believe that the partial shape recognition problem can be effectively attacked using the same procedure.

REFERENCES

[1] G. H. Granlund, "Fourier Preprocessing for Hand Print Character Recognition," IEEE Trans. on Computers, Vol. C-21, pp. 195-201, Feb, 1972.

[2] E. Persoon and K. S. Fu, "Sequential Decision Procedures with Prespecified Error Probabilities and Their Applications," School of Electric. Eng., Purdue Univ., West Lafayette, IN, Tech. Rep. TR-EE 74-30, 1974.

[3] E. Persoon and K. S. Fu, "Shape Discrimination Using Fourier Descriptors," IEEE Trans. Syst., Man, Cybern., Vol. SMC-7, pp. 170-179, March, 1977.

[4] S. A. Dudani et al, "Aircraft Identification by Moment Invariants," IEEE Trans. on Computers, Vol. C-26, pp. 39-46, Jan, 1977.

[5] C. W. Richard, Jr., and H. Hemami, "Identification of Three-Dimensional Objects Using Fourier Descriptors of the Boundary Curve," IEEE Trans. Syst., Man, Cybern., Vol. SMC-4, pp. 371-378, July, 1974

[6] T. Wallace and P. A. Wintz, "Fourier Descriptors for Extraction of Shape Information," Final Report of Research for the Period Nov. 1, 1975 -Oct. 31, 1976, ARPA Contract No. F 30602-75-C-0150.

[7] U. Montanari, "A note on Minimal Length Polygonal Approximation to a Digitized Contour", Comm. ACM, Vol 13, pp. 41-47, Jan., 1970.

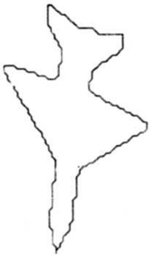
[8] R. Brons, "Linguistic Methods for the Description of a Straight Line on a Grid," Computer Graphics and Image Processing, Vol. 3, No. 1, pp. 48-62, March, 1974.

[9] A. Rosenfeld, "Digital Straight Line Segments," IEEE Trans. on Computers, Vol. C-23, pp. 1264-1269, December 1974.

[10] H. Freeman, "Application of the Generalized Chain Coding Scheme to Map Data Processing," in Proceedings of the 1978 IEEE Computer Society Conference on Pattern Recognition and Image Processing, May 1978.

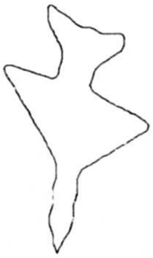
[11] T. Wallace and P. A. Wintz, "Three Dimensional Aircraft Recognition Using Fourier Descriptors," Proceedings: Image Understanding Workshop, October 1977, (Science Applications Inc. Report No. SAI-78-656-WA).

[12] K. S. Fu, "Syntactic Methods in Pattern Recognition," Academic Press, New York, 1974.



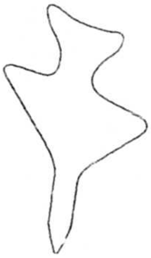
No Filtering

Fig. 2



2 % Rectangular

Fig. 3



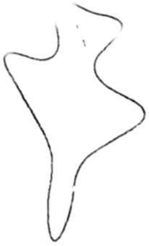
4% Rectangular

4

Chain Code Error

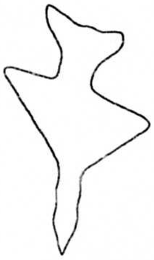
Uniform Sampling

Fig. 1



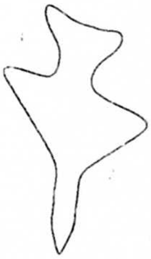
6 % Rectangular

Fig. 5



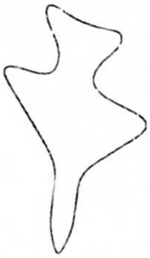
4 % Triangular

Fig. 6



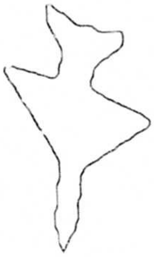
6 % Triangular

Fig. 7



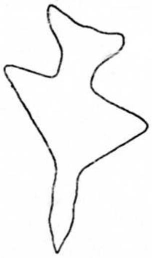
8 % Triangular

Fig. 8



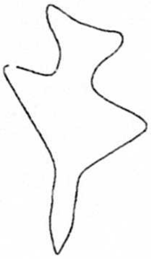
3 % Gaussian

Fig. 9



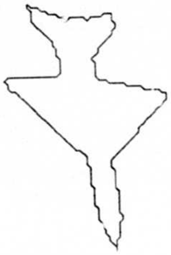
5 % Gaussian

Fig. 10



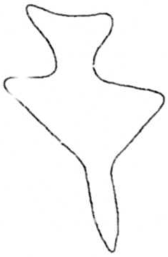
7 % Gaussian

Fig. 11



No Filtering

Fig. 12



4 % Rectangular

Fig. 13

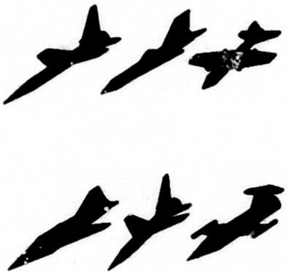


Fig. 14

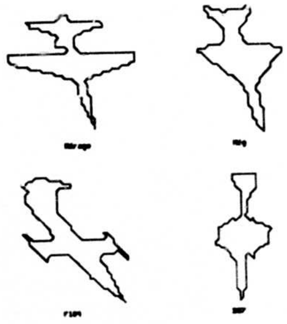


Fig. 16

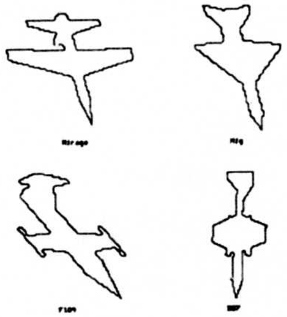


Fig. 15

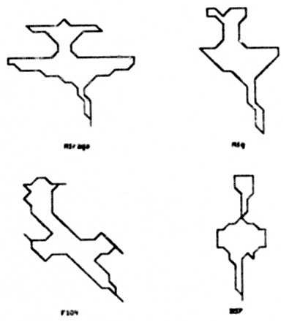


Fig. 17

CLASSIFICATION ACCURACY
VS. WINDOW CHARACTERISTICS

Width	Rectangular	Triangular	Gaussian
2 %	86.7 %	84.3 %	
3 %	86.7 %		86.0 %
4 %	88.0 %	85.3 %	87.0 %
5 %	86.3 %	85.3 %	87.3 %
6 %	84.7 %	87.3 %	
7 %		86.3 %	86.7 %
8 %		85.7 %	
10 %			85.7 %

The window used for digital filtering was varied and the classification accurccy shown was achieved for data set 2, with parameters k = 10, d = 2.0, and absolute value distance measure.

Table 1

CLASSIFICATION ACCURACY VS. NORMALIZATION METHOD

Approx.	k	d	METHOD 1		METHOD 2	
S/N			M. S.	ABS. VAL.	M. S.	ABS. VAL.
41 dB	1	-	81.0 %	83.7 %	83.0 %	84.3 %
41 dB	10	2.0	82.7 %	84.3 %	85.3 %	88.0 %

Normalization method 1 used the phase of a single coefficient to resolve the normalization ambiguity, whereas method 2 optimized the sum of the "positive real energy" as explained in the text. The results shown are for data set 2, and the absolute value distance measure. The other tables show performance using normalization method 2 exclusively.

Table 2

CLASSIFICATION ACCURACY VS. SIGNAL TO QUANTIZING NOISE RATIO (NO DIGITAL FILTERING)

Approx.	k	d	DATA 1		DATA 2	
S/N			M. S.	ABS. VAL.	M. S.	ABS. VAL.
41 dB	1	-	92.3 %	92.3 %	84.7 %	84.0 %
41 dB	10	2.0	92.7 %	92.3 %	86.7 %	87.7 %
35 dB	1	-	83.3 %	87.0 %	83.7 %	82.0 %
35 dB	10	2.0	84.7 %	86.7 %	83.7 %	84.3 %
29 dB	1	-	69.0 %	68.7 %	67.0 %	67.0 %
29 dB	10	2.0	69.7 %	68.7 %	68.3 %	67.7 %

CLASSIFICATION ACCURACY VS. SIGNAL TO QUANTIZING NOISE RATIO (4 % RECTANGULAR WINDOW FILTERING)

Approx.	k	d	DATA 1		DATA 2	
S/N			M. S.	ABS. VAL.	M. S.	ABS. VAL.
41 dB	1	-	91.0 %	93.0 %	83.0 %	84.3 %
41 dB	10	2.0	93.0 %	94.7 %	85.3 %	88.0 %
35 dB	1	-	86.0 %	86.7 %	82.3 %	84.0 %
35 dB	10	2.0	88.3 %	89.0 %	84.3 %	85.3 %
29 dB	1	-	69.3 %	72.3 %	63.3 %	62.0 %
29 dB	10	2.0	69.7 %	71.0 %	65.7 %	64.0 %

The resolution of the library data was 256 x 256 (47 dB) in each case. The nearest "k" projections to each unknown projection were investigated, unless their distance was more than "d" times the minimum distance.

Table 3

ESTIMATION ACCURACY
(4 % RECTANGULAR WINDOW FILTERING)

Approx. S/N	k	d	MEDIAN ANGLE ERROR			
			M. S.		ABS. VAL.	
			x	y	x	y
Data 1						
41 dB	1	-	.0597	.0343	.0545	.0371
41 dB	10	2.0	.0569	.0315	.0543	.0315
Data 2						
41 dB	1	-	.0651	.0477	.0578	.0478
41 dB	10	2.0	.0556	.0477	.0556	.0478

The estimation algorithm results in much smaller angle errors than can be achieved by methods which simply assume the unknown projection to be oriented in the same way as the nearest library projection. The angle errors which would be expected using conventional methods, assuming that each correct classification is made using the correct nearest library projection, and using our density of projections, are about .12 radians for the x resolution, and .14 radians for the y. These numoers represent an upper bound for a conventional system.

Table 4

CLASSIFICATION ACCURACY FOR UNKNOWN PROJECTIONS
TAKEN FROM COMPLETELY ARBITRARY DIRECTIONS
(4 % RECTANGULAR WINDOW FILTERING)

Approx. S/N	k	d	CLASSIFICATION ACCURACY	
			M. S.	ABS. VAL.
41 dB	1	-	81.5 %	83.2 %
41 dB	10	2.0	84.7 %	84.7 %

Although this algorithm is only designed to recognize aircraft viewed from above, and the library of projections only contains projections taken from above, a random set of 600 projections taken half from above and half from below was tested with the results shown.

Table 5

SYNTACTIC RECOGNITION OF TACTICAL TARGETS

Raj K. Aggarwal & Timothy M. Wittenburg

Honeywell, Inc.
2600 Ridgway Parkway
Minneapolis, Minnesota 55413

ABSTRACT

A syntactic scheme for tank-truck-clutter recognition in FLIR images is described here. It involves seven steps: coarse segmentation, detailed segmentation, supergroup formation, initial classification, vertical truck recognition, horizontal truck recognition and tank recognition. Prototype similarity transformation [1] is used to perform coarse and detailed segmentations. Experimental results on FLIR images containing tactical targets are included.

INTRODUCTION

In picture recognition problems, the number of features required for statistical pattern recognition is often very large, which makes the idea of describing complex patterns in terms of a (hierarchical) composition of simpler subpatterns very attractive [2] . Also, if the number of possible descriptions is very large as is the case for tactical targets from relatively close range, it is impractical to regard each description as defining a class. Consequently, the requirement of recognition can be better satisfied by a description of each class rather than by its classification.

For example, consider the image of a tank shown in Figure 1. Suppose it is possible to recognize the component parts of this tank as motor, hot vents, barrel, etc., using statistical properties of each component and their spatial relationship. The hierarchial (tree-like) structural information in this tank can be represented by a tree as shown in Figure 2. Grammatical rules can then be used to describe these trees. The grammatical rules for this example are:

TANK → RECTANGLE, HOTSPOTS, BARREL

RECTANGLE → TREAD, MOTOR, VENTS

Since different components of a target may be seen from different aspect angles, a general set of rules can be infered by training the classifier with tree-structures of the target viewed from different aspect angles. The general block diagram of syntactic approach to tactical target recognition is shown in Figure 3.

The assumption in this approach to tactical target recognition are:

- Images of tactical targets are "large" enough to show structure.

- It is easier to recognize target components than the target.

The first assumption deals with the sensor-target range. If the range is too large to show any details inside the target, one would have to resort to statistical recognition techniques. But as the sensor-target range decreases and the target structure becomes discernable, syntactic recognition schemes become feasible. From our experience, if the target area is of the order of one-half to one percent of sensor FOV, syntactic recognition schemes are feasible. This translates to about a ten centimeter pixel resolution.

The second assumption deals with the relative ease of recognizing target and its components. If it is easier to recognize a target than its components as would be the case when target image is only a few pixels, one would not employ syntactic recognition schemes. But in low quality images where the recognition based on target outline is not very reliable, a snytactic scheme can be successfully used to recognize targets provided the assumption on target image size holds. Even for good quality images, target orientations will result in different target outlines. Consequently, one will need several statistical classifiers for each type of target. In principle, one set of syntactic rules can be generated to recognize the target from all aspect angles. Syntactic recognition schemes can also be successfully used for partially occluded targets where conceivably statistical recognition schemes would fail.

In the following sections, a version of a syntactic scheme to perform tank-truck-clutter recognition in FLIR images is described. Experimental results on FLIR images containing tanks and trucks are also included.

TANK-TRUCK-CLUTTER RECOGNITION IN FLIR IMAGES

A syntactic scheme for recognizing tanks and trucks and discriminate them against clutter is described in the following seven steps:

- Coarse Segmentation
- Detailed Segmentation
- Supergroup Formation
- Initial Classification
- Vertical Truck Recognition
- Horizontal Truck Recognition
- Tank Recognition

Coarse Segmentation

The objective of this step is to isolate potential targets in a full frame. The block diagram describing this step is shown in Figure 4. Prototype similarity transformation [1] is applied. on a full frame at a low resolution using the intensity attribute for initial coarse segmentation. A labeling program is then used to form groups. A group is defined here as a collection of adjacent cells having the same symbolic representation. A label table is formed containing symbolic information about each group. The symbolic information vector consists of size, shape, position features. This label table is then used to form supergroups. A supergroup is defined as groups within a maximum distance of IDIS. IDIS is set proportional to the super group size to keep supergroup formation independent of the sensor-target distance. If a supergroup consists of only one group and is smaller than a minimum size or if it touches the edges of the frame, it is removed from further consideration. Each remaining supergroup is enclosed by a subframe of a suitable size. Overlapping subframes are combined to form one subframe. The subframes are then processed sequentially through the remaining steps.

Detailed Segmentation

The objective of this step is to isolate components of a target in the subframe. The block diagram describing this step is shown in Figure 5. Detailed segmentation is performed by an iterative use of prototype similarity transformation at a smaller cell size to permit fine resolution. The edge and intensity attributes are used to supply additional detail to the segmentation. Recall that coarse segmentation was performed using the intensity attribute only. The output of this step is a symbolic image of the subframe.

Supergroup Formation

The objective of this step is to retain only those components which may be part of a single target. The block diagram describing this step is shown in Figure 6. Same scheme is used for the supergroup formation as is described in the coarse segmentation step. Symbolic information table for the supergroup is passed on to the initial classification step.

Initial Classification

The objective of this step is to classify the object initially into a possible horizontal truck, vertical truck or a tank based on assumed models of a truck and a tank. The block diagram describing this step is shown in Figure 7. Approximate orientation of the supergroup is determined by noting whether the supergroup extent in the X-direction exceeds that in the Y-direction or vice versa. Assuming the target type group with the largest area, say A, to be either body (for a tank) or box (for a truck), search regions based on the area A are established as shown in Figure 7. An attempt is made to find a cab (for a truck) or a motor (for a tank) using size and shape features in search regions I and II. If no additional component is found in either of the search regions, the object is classified as a possible horizontal truck. If the group found in either search region is not totally enclosed by an edge group, the object is classified as a possible vertical truck. Otherwise the initial classification is a possible tank.

Vertical Truck Recognition

The objective of this step is to utilize the label table of a possible vertical truck and classify it as a vertical truck or a possible horizontal truck. The block diagram describing vertical truck recognition is shown in Figure 8. If components are found in both search regions in initial classification step, the possibility of a three component truck is eliminated. If a component is found in only one search region and is classified as a cab of the truck using size and shape features in the initial classification step, an attempt is made to find a motor in front of the cab using size and shape features. If a motor is found, the object is classified as a vertical truck. If components are found in both search regions or no motor is found, a test using relative component size is made to determine if the object is possibly a two-component truck. Otherwise the object is tested for a horizontal truck.

Horizontal Truck Recognition

The objective of this step is to utilize the label table of a possible horizontal truck and classify the object as a horizontal truck or clutter. The block diagram for horizontal truck recognition is shown in Figure 9. A "step" detector program is used to detect and locate "steps" in the box group identified in initital classification step. If no "steps" are detected, the object is classified as clutter. If "steps" are detected, additional components are defined at the "step" junctions. An attempt is made to find a box using shape features. If the attempt fails, the object is classified as clutter. Otherwise, the object is classifed as a horizontal truck. Size and shpae features are used to test for the presence of additional components such as cab and motor of the truck.

Tank Recognition

The objective of this step is to utilize the label table for a possible tank and decide whether the object can be classified as a tank based on identification of the additional components (motor has already been identified at the initial classification step). If no additional tank components can be identified, boundary shape analysis is required for recognition. The block diagram for tank recognition is shown in Figure 10. An approximate orientation of the object is determined using the already defined body and motor groups. Direction of the possible tank is determined by examining the location of the motor group relative to the body group. Search regions are established for locating hot spots, vents and barrel of the tank as shown in Figure 10. Size, shape and direction features are used for component recogntion. If at least one additional tank component is found, the object is declared a tank. Otherwise, statistical methods based on object boundary features are needed for further classification.

EXPERIMENTAL RESULTS

The technique for tank-truck-clutter recognition was applied to FLIR images containing tactical targets. The coarse segmentation was done on full frames (520 pels x 480 pels). The target center and its approximate size were recorded during digitization. The 8 bit digitized data was scaled down to 100 gray levels to reduce computer memory requirements for storing the joint distribution function.

A cell was defined as 4 pels x 4 pels for coarse segmentation and 2 pels x 2 pels for detailed segmentation. A neighborhood of 3 cells x 3 cells was used in both cases for calculating the joint distribution function. The intensity and edge attributes were used for detailed segmentation. A threshold of 0.85 was used to define the prototype intervals. Approximately the same number of prototypes (∿10) were obtained in both cases. Target and background cues for edge attribute were located at fixed percentiles in the edge attribute histogram. The intensity of the target center recorded during digitization was taken as intensity target cue and intensity at cell (1,1) was taken as the background cue.

Two shape features, length/width and diffuseness = area/(length x width) were used for component recognition.

The results are shown in Figures 11-13. In each set of the five photographs, the original picture, coarse segmentation, false colored image after the supergroup formation, component recognition and the object recognition are shown starting from top left.

Initial testing on a limited data set of eight tanks, eight trucks and six false alarms resulted in perfect discrimination.

DISCUSSION

Initial experimental results demonstrate the feasibility of using a recognition scheme based on syntactic techniques for tank-truck-clutter recognition in FLIR images. However, there are a few points worth noting. Firstly, before a syntactic recognition scheme can be utilized, the two assumptions noted earlier, namely; objects are large enough to show their structure and target components are easier to recognize than the target, should hold. Secondly, this scheme has been developed for recognizing assumed models of tanks and trucks. However, the restriction on the models are mainly due to our limited data set. Thirdly, since we don't have data showing targets in all possible orientations, our assumed tank and truck models are very simple. Consequentally, a formal grammer has not been developed.

ACKNOWLEDGEMENTS

The authors wish to thank Dr. M. Geokezas, the Section Chief of Signal and Image Processing at Honeywell for helpful discussions and encouagement. They also wish to thank Ms. Karyn Huddleson for skillful typing in preparing this manuscript.

REFERENCES

1. Aggarwal, R. K., Image Segmentation for Syntactic Classification of Large Images. Image Understanding Workshop Proceedings, Boston, page 125-132, May 1978.

2. Fu, K. S., Syntactic Methods in Pattern Recognition, Academic Press, 1974.

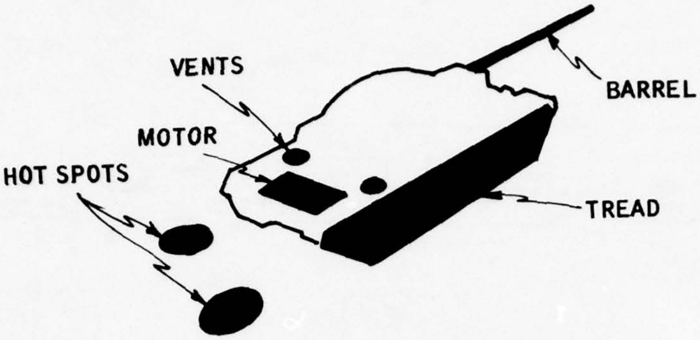


Figure 1. Image of a Tank.

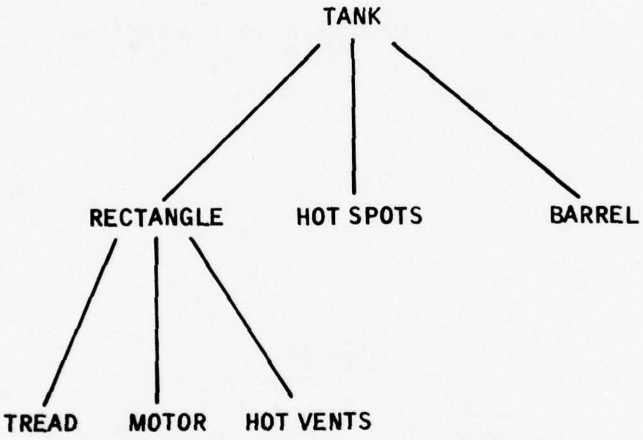


Figure 2. Heirarchical Structural Description of the Tank Shown in Figure 1.

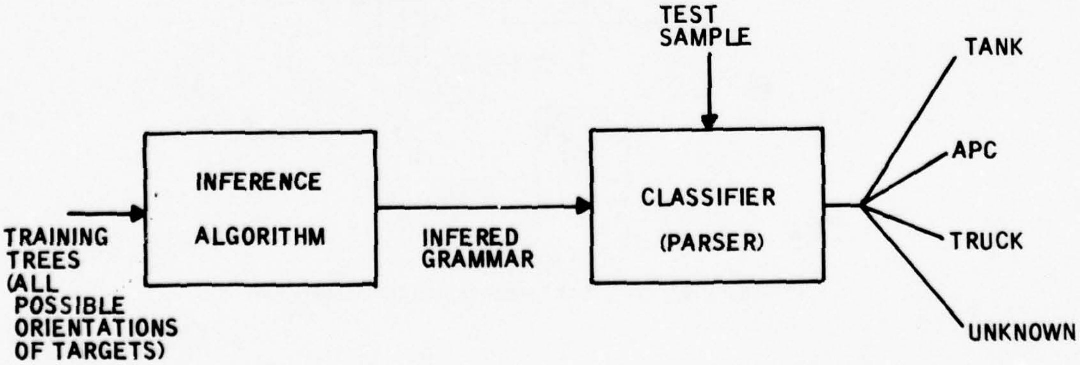


Figure 3. Syntactic Approach for Tactical Target Recognition.

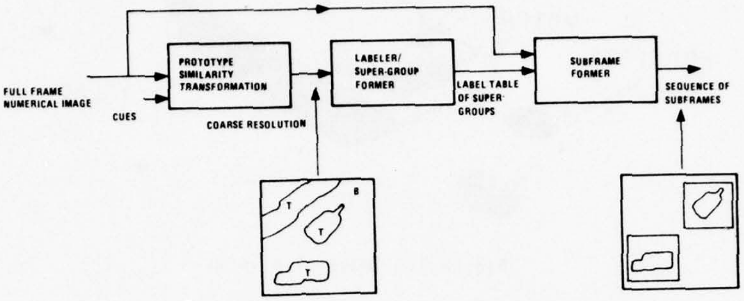


Figure 4. Coarse Segmentation in a Full Frame.

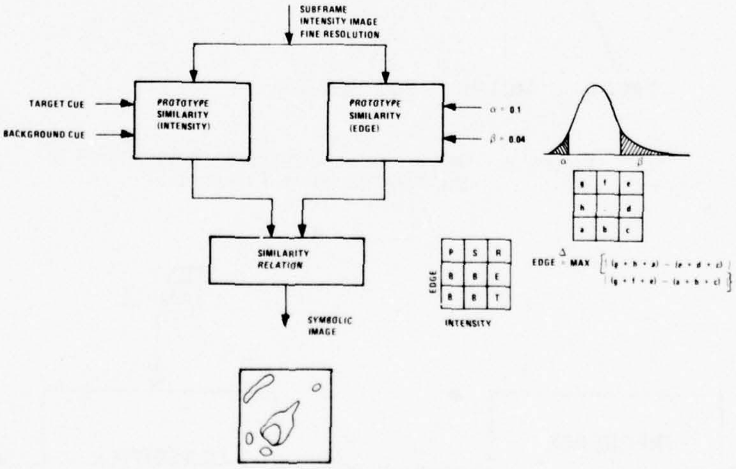


Figure 5. Detailed Segmentation in a Subframe.

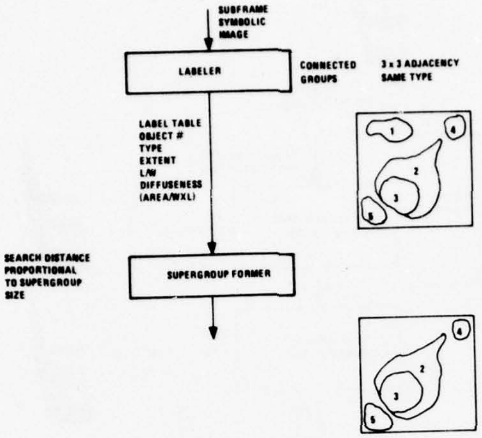


Figure 6. Supergroup Formation in the Subframe.

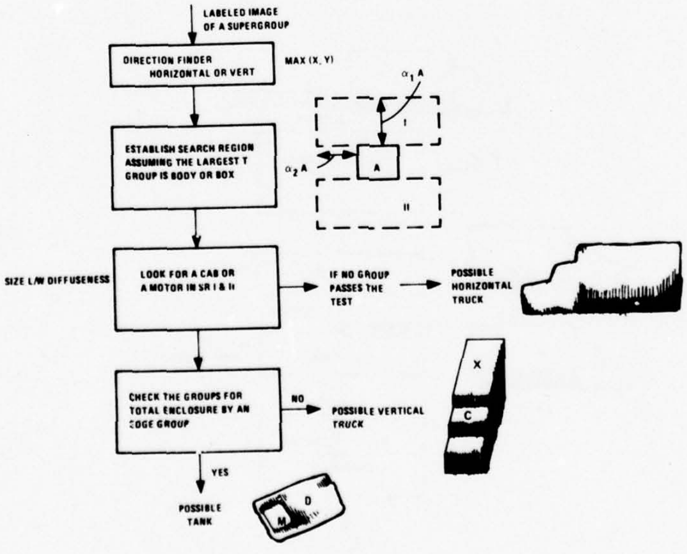


Figure 7. Initial Classification.

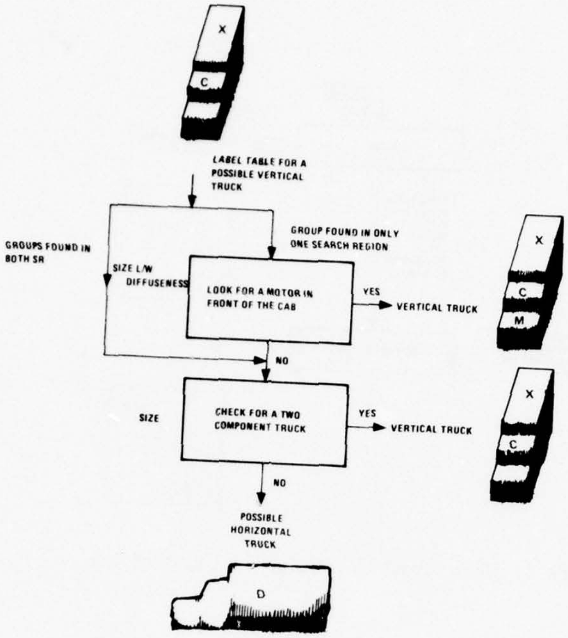

X
C
LABEL TABLE FOR A POSSIBLE VERTICAL TRUCK
GROUPS FOUND IN BOTH SR
GROUP FOUND IN ONLY ONE SEARCH REGION
SIZE L/W DIFFUSENESS
LOOK FOR A MOTOR IN FRONT OF THE CAB
YES
VERTICAL TRUCK
X
C
M
NO
SIZE
CHECK FOR A TWO COMPONENT TRUCK
YES
VERTICAL TRUCK
X
C
NO
POSSIBLE HORIZONTAL TRUCK
D

Figure 8. Vertical Truck Recognition.

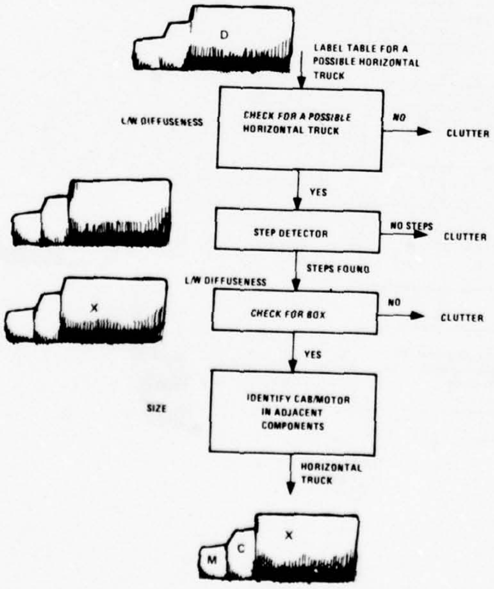


Figure 9. Horizontal Truck Recognition.

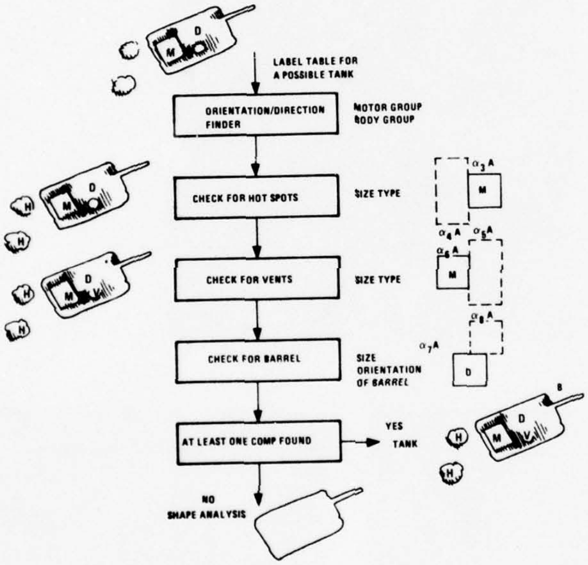


Figure 10. Tank Recognition.

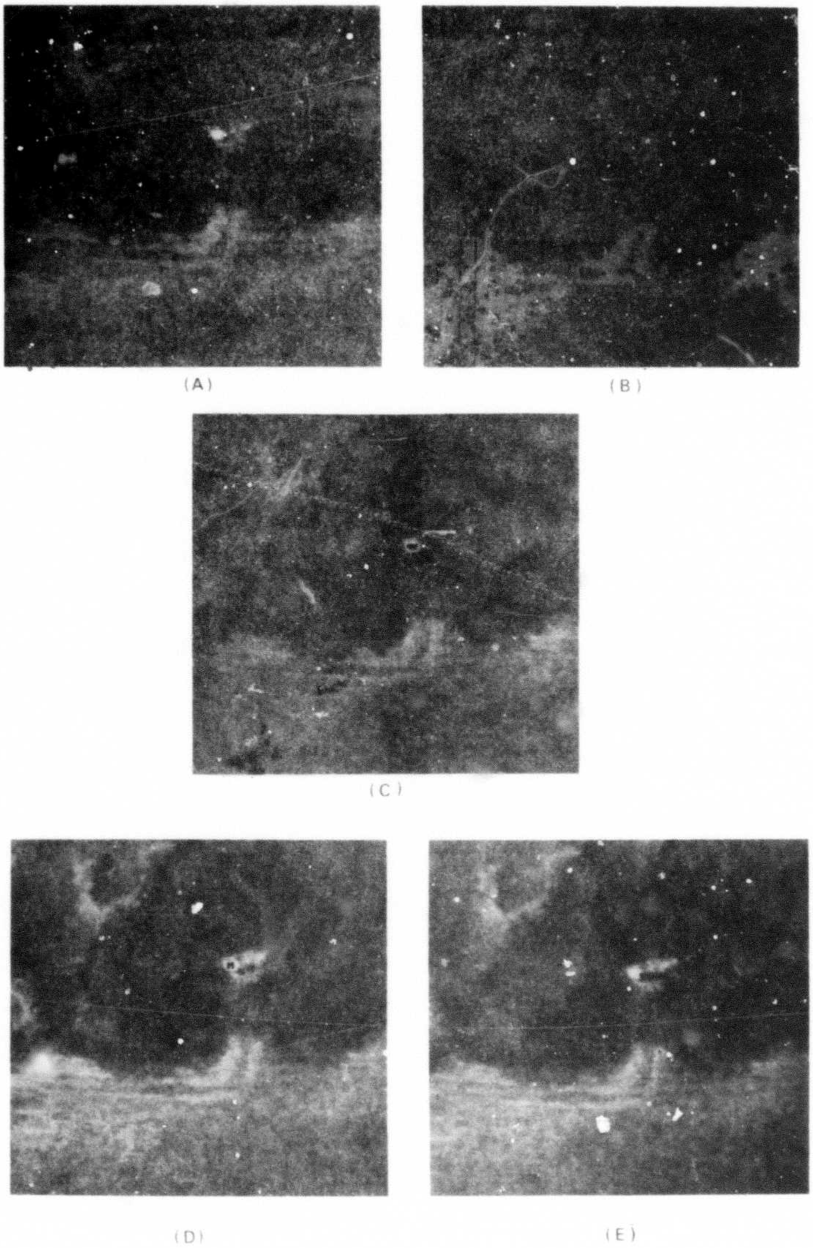


Figure 11. (A) FLIR Image of a Scene Containing a Tank.
(B) Coarse Segmentation.
(C) False Color Image After Supergroup Formation.
(D) Component Recognition.
(D) Target Recognition.

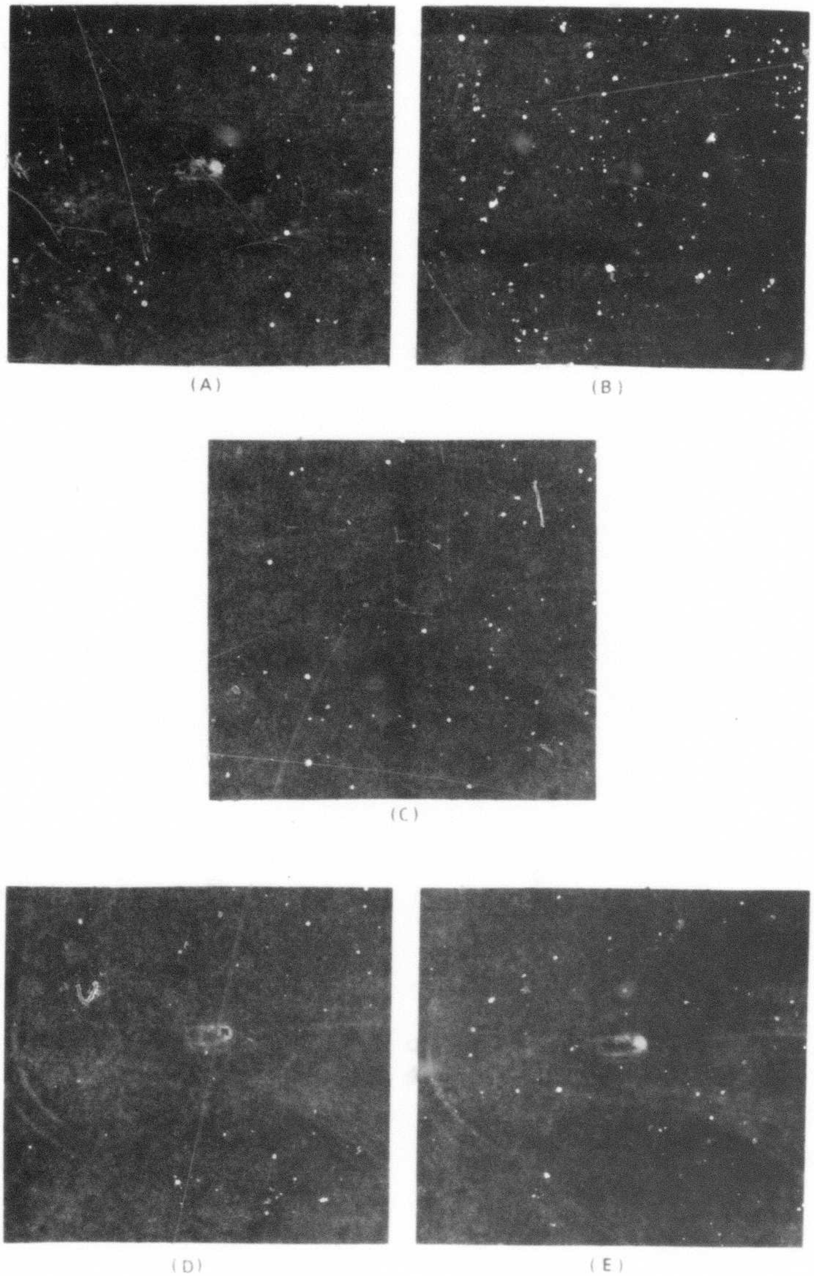


Figure 12. (A) FLIR Image of a Scene Containing a Tank.
(B) Coarse Segmentation.
(C) False Color Image After Supergroup Formation
(D) Component Recognition
(E) Target Recognition

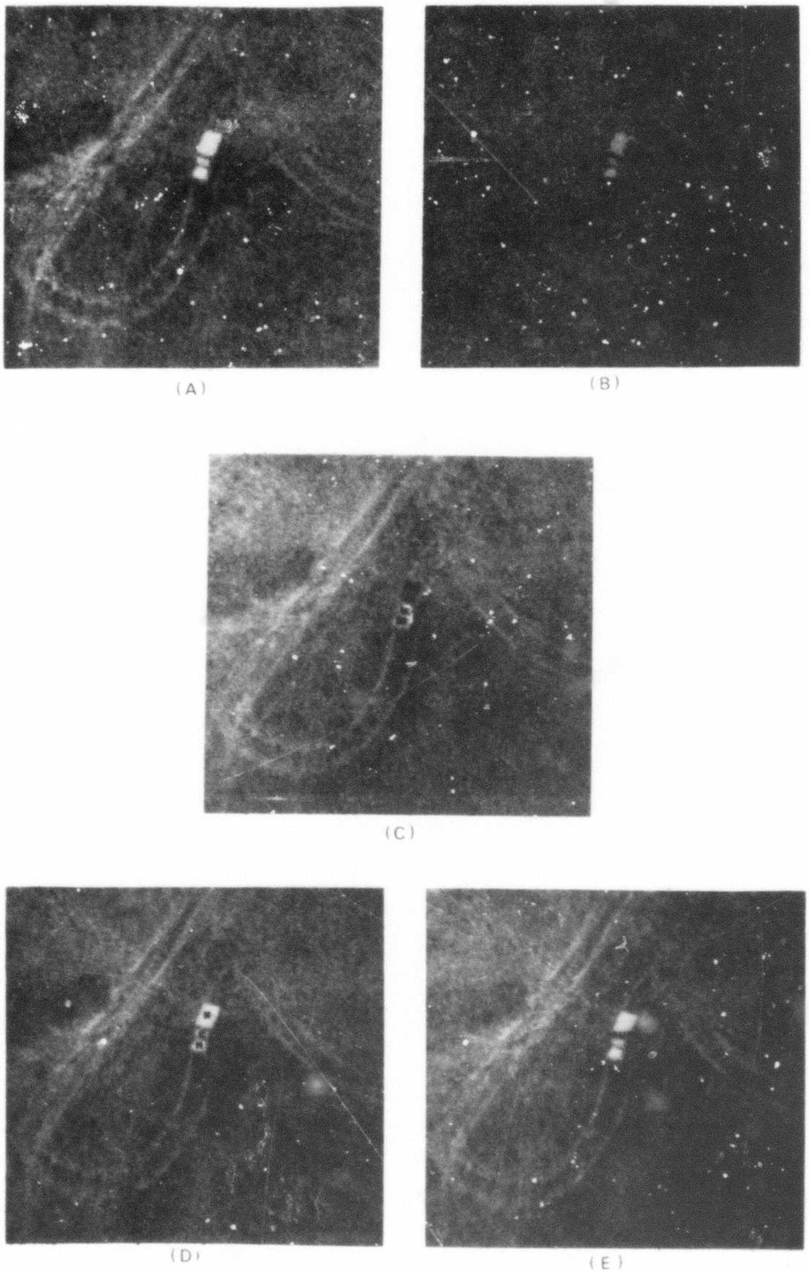


Figure 13. (A) FLIR Image of a Scene Containing a Truck.
(B) Coarse Segmentation.
(C) False Color Image After Supergroup Formation.
(D) Component Recognition.
(E) Target Recognition.

SEGMENTATION AND CLASSIFICATION OF TARGETS IN FLIR IMAGERY

O. Robert Mitchell and Stephen M. Lutton
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

An approach is described for detecting and classifying tactical targets in FLIR imagery. The basic assumption used for segmenting objects from their background is that the objects to be detected differ from the background in grey level, edge properties, or texture. Potential targets are selected from a large frame by locating combinations of grey level, edge value, and texture that occur infrequently over the entire frame. Once potential objects are obtained, they are segmented from their backgrounds using the identical process as above, except appiied on a local level. The segmented objects are classified into three types of vehicles or into false alarms. The classification procedure uses features measured on projections made through the segmented objects. Results are shown for 32 test images.

1. Introduction

The problem being considered in this paper is the automatic detection and classification of tactical targets in forward looking infrared (FLIR) imagery. Typical images are shown in Figs. 1 and 2. The camera is similar to a television camera but the sensor is sensitive to radient thermal emission instead of visible light. The thermal images produced (white is hot, black is cold) tend to lack the sharpness of higher frequency imagery. The ultimate goal is to implement this classification system in real-time. However, this paper discusses algorithms for detection and classification of such objects from a single frame, independent of the real-time constraints.

The problem is divided into three sections: (1) selection of potential object locations; (2) segmentation of these objects from their background; and (3) classification of the segmented objects.

2. Selection of Potential Target Locations

The assumption made in this section is that combinations of grey level, edge value, and texture that occur only a few times over an entire frame of imagery are potential target points. As an example, consider Fig. 1. An edge picture is generated using a smoothed gradient measured over a 7x7 window at each point. The absolute difference between the upper 21 points and the lower 21 points is compared against the absolute difference between the left 21 points and the right 21 points. The center point is then replaced by the maximum of these two values. This process is repeated for each point in the original image to produce the edge feature image. The resulting edge feature image for Fig. 1 is shown in Fig. 3.

The texture feature is derived from the max-min local extrema described previously [1,2]. Local grey level extrema are measured in hysteresis smoothed versions of the original image using three smoothing thresholds. The lowest level extrema correspond mostly to noise in the image, whereas the highest correspond mostly to edges. The remaining medium level extrema are a measure primarily of the texture in the image. These medium level extrema locations are shown in Fig. 4. The texture feature image in Fig. 5 is created from the extrema by averaging the number of medium level extreme in every 10x10 window in the image and replacing the center point of that window with the average.

Once the three feature images (grey level, edge, and texture) are available, a three-dimensional histogram is generated for the frame using a quantization of 32 grey levels, 16 texture values, and 8 edge values. This histogram is therefore composed of 4096 bins. Points in the original image are then located whose three value combination occurs infrequently. Shown in Fig. 6 are all such locations having a combination occurring less than 15 times in the entire image. The location of potential targets is then made by finding concentrated clusters of such points.

The same process is repeated for Fig. 2 and the resulting potential target points are shown in Fig. 7.

3. Segmentation of Potential Targets

Once a potential target's location is known, the target must be segmented from its background as accurately as possible. This is done by collecting local statistics from the background immediately surrounding the object and finding all points in the target region which do not match the background. Two composites, each showing 16 potential targets are shown in Figs. 8 and 9. The resulting edge and texture feature images for

Fig. 8 are shown in Figs. 9 and 10.

Information from the potential target location system described earlier provides the approximate target size. The three dimensional histogram is collected from an annular region surrounding the potential target. For the composites shown in Figs. 10 and 11, the inner radius was 35 and the outer radius was 64.

Once the background 3-D histogram is completed, each potential target point (3-D vector) is compared against its background bin. If that feature combination occurs often in the background, the point is considered another background point. If the feature combination does not occur in the background, that point is labeled a target point. The threshold used in the examples shown here was 3 occurrences of a particular grey level, edge and texture in the background was sufficient to remove potential target points having that same combination. All points in the original image matching its local background are set to zero (black) giving the segmented results shown in Figs. 12 and 13.

Although the segmentations produced have flaws, the only processing necessary prior to classification is the application of a 3x3 median filter. This cleans up isolated points and holes and gives better projection data for the classification which follows.

4. Classification of the Segmented Objects

The segmentations produced by the method previously described produce results which are sometimes fragmented and contain drop-out and extraneous points. A classification scheme which is somewhat insensitive to these variations would be appropriate. We are presently investigating the use of projections through the segmented object to derive classification features. A similar type of structure recognition method is being developed by New Mexico State University for missile tracking at the White Sands Missile Range [3]. It has the advantage that the integration process of the projections averages out many of the noise problems inherent in thermal images and our segmentation method.

Projections are produced by summing grey levels in the segmented pictures along parallel straight lines. We have used 8 directions of projections. Projection number 0 corresponds to vertical summation lines; projection 1 corresponds to lines oriented at +22 1/2 degrees with respect to vertical; and continuing in 22 1/2 degree increments to projection 7 which corresponds to +157 1/2 degrees with respect to vertical.

Only the widest and narrowest of these eight projections are retained. The width of a projection is found by measuring the distance between 20% of the total area and 80% of the total area under the projections. The resulting projections for all 32 potential targets are shown in Fig. 14.

The classification features are derived from distinguishing characteristics expected due to the nature of the four types of objects. These characteristics are:

(1) Tank - The motor predominates as a hot spot: this results in a dominant peak in the wide projection and a peaked middle in the narrow projection. Vent holes in front of the motor show as a dark region resulting in a dip in the wide projection near the center causing the narrow projection to be even more peaked.

(2) APC - The armoured personnel carrier has a smaller and less dominating motor and a seating area near the center. The wide projection is fairly symmetrical with a dip in the center. The narrow projection is more square than the tank.

(3) Truck - The windshield usually shows as a dip in the wide projection near one end of the projection. The change from body to hood usually appears as a decrease in the wide projection. The narrow projection is usually more peaked than the APC but less than the tank.

(4) False Alarm - The false alarms may give a highly varying projection, one that is inconsistent with the targets (too wide, too narrow in one projection or the other). Also any projection which does not fall into one of the target categories is classified as a false alarm.

The following classification procedure is used for the projections:

(1) Calculate the width of the narrow and wide projections (NW and WW). If $\frac{WW}{NW} > 2.1$ or if either projection extends off one edge, the object is classified as a false alarm. [Examples: FA88, FA111, FA112, FA113, FA115, FA124, and Tank175].

(2) Generate the hysteresis smoothed local maxima and minima [1] along each projection. The presence of more than two large maxima in any projection implies a flase alarm. [Examples: FA112, FA124, Tank175].

(3) Look for projections with two dominant peaks on each side of the middle of the wide projection with a valley located approximately in the center. If the dominant peak height is DPH and the second peak height is SPH and the valley between them is VH, and if

$$\frac{DPH - VH}{SPH - VH} < 1.7 \text{ and } \frac{SPH - VH}{SPH} > .1 \qquad (1)$$

the target class is APC. [Examples: APC25, APC28, APC96, APC98, APC132].

(4) If

$$\frac{DPH - VH}{SPH - VH} > 1.7 \text{ and } \frac{SPH - VH}{SPH} > .1 \qquad (2)$$

there is one dominating peak and the target is a tank or a truck. Distinction is made based on the location of the valley (VL) relative to the second peak location (SPL) and the dominant peak location (DPL). If there are two small, approximately equal valleys between the dominant and second peaks, use the one closest to the second peak (truck windshields sometime do this).

If

$$.8 < \frac{DPL - VL}{SPL - VL} < 1.3 \qquad (3)$$

classify the target as a tank[Examples: Tank3, Tank4, Tank5s, Tank8, Tank92, and Tank4].

If

$$1.3 < \frac{DPL - VL}{SPL - VL} < 3.0 \qquad (4)$$

classify the target as a truck. [Examples: Tank14, Truck69, Truck71, Truck72, Truck144s, and Truck145s].

(5) If

$$\frac{DPH - VH}{SPH - VH} > 1.7 \text{ and } \frac{SPH - VH}{SPH} < .1 \qquad (5)$$

there is only one dominating peak and no definite valley. Classify the object as a truck. [Example: Truck58].

(6) If

$$\frac{DPH - VH}{SPH - VH} < 1.7 \text{ and } \frac{SPH - VH}{SPH} < ,1 \qquad (6)$$

or there is no dominant valley and the projection is approximately equal on both sides, the class is either APC or truck. Discrimination is made from the peakiness of the narrow projection. The parameter measured is the width of the top 20% of the peak relative to the total width. If this is < .40, classify as truck. [Examples: Truck34 and Truck 38]. If the relative width is > .40, classify as APC. [Examples: APC30, APC55, APC142] .

(7) Any other objects not yet classified are called false alarms. [Examples: FA86 (two major peaks on same side of center line) and FA125 (one dominant peak located in center).

Results of Classification

Most objects were correctly classified using the above procedure. Of the two misclassifications, Tank17s was called a false alarm and Tank14 was called a truck. The former was caused by poor segmentation and the latter by occlusion of part of the tank by woods. The results are very encouraging.

5. Conclusions

It is possible to locate potential targets and classify them with good accuracy using grey level, texture, and edge measurements and projections through the segmented results. Of course the limited data sample here prevents the certainty of extension of this technique to all FLIR sensors, aspect angles, times of day, targets, and backgrounds. However the robust nature of both the segmentor and the classifier should allow a wide variation in data with good performance.

The question of real-time implementation has not been addressed here. Certainly use could be made of past frame information, so that background statistics are already available for use on the next successive frame. Also classifications can be repeated over successive frames for more accurresults.

Although features from the projections were chosen that were generic to the types of targets being classified, improvement might be further gained by optimizing the feature selection over a larger set of data.

ACKNOWLEDGEMENTS

We wish to acknowledge the efforts of Steve Carlton and Robert Ward, who provided the early results and programs on which this work was based. We also would like to acknowledge the pioneering work of G. M. Flachs at New Mexico State University and A. L. Gilbert at White Sands Missile Range who have provided the concepts of background elimination and classification by projections for real-time tracking purposes.

REFERENCES

1. Mitchell, O.R., Myers, C.R., and Boyne, W.A., "A Max-Min Measure for Image Texture Analysis," IEEE Transactions on Computer, Vol. C-26, pp. 408-414, April 1977.

2. Carlton, S. G. and Mitchell, O. R., "Image Segmentation Using Textures and Gray Level," Proceedings of the IEEE Conference on Image Processing and Pattern Recognition, Troy, N.Y., pp. 387-391, June 6-8, 1977.

3. Flachs, G. M., Thompson, W. E., and Yee-Hsuun U, "A Real-Time Structural Tracking Algorithm," NAECON 1976 Record, pp. 161-168.

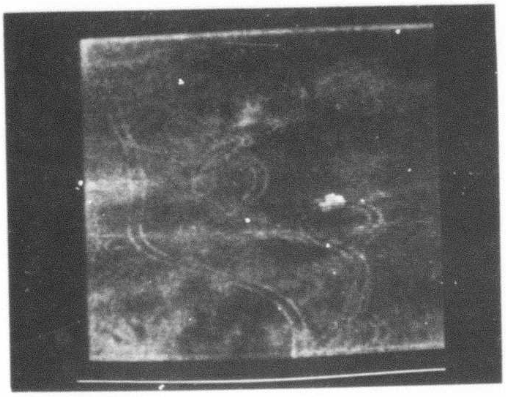


Fig. 1. Original 8 bit image. The picture size is 500x480 pixels. The target is a tank.

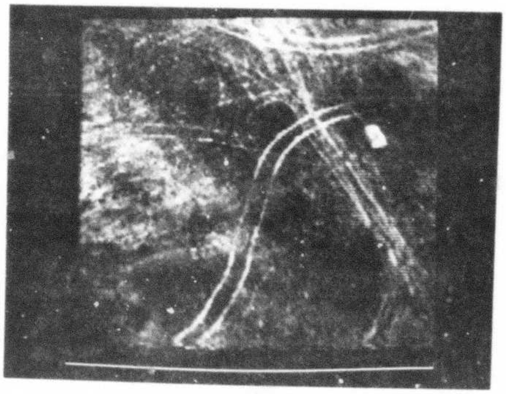


Fig. 2. Original 8 bit image. The picture size is 500x480 pixels. The target is an APC.

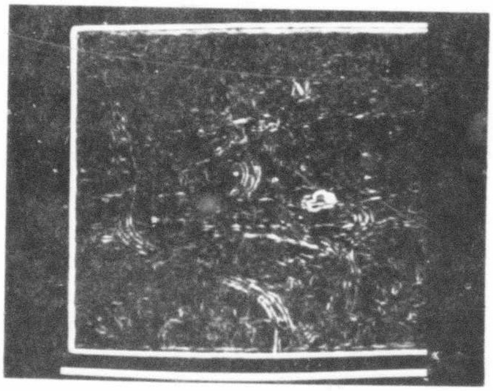


Fig. 3. The resulting edge feature image for Fig. 1 using a smoothed gradient measure over a 7x7 window.

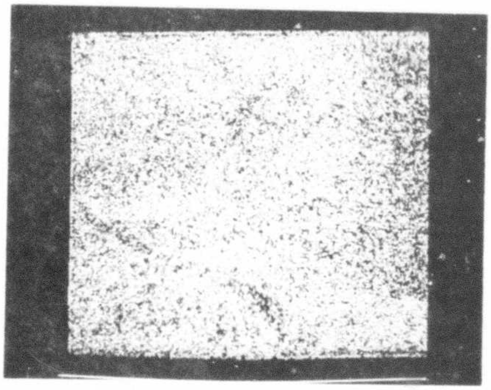


Fig. 4. Medium level local grey level extrema present in Fig. 1.

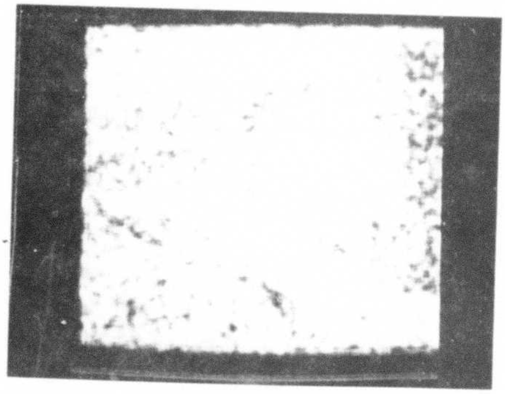


Fig. 5. Texture feature formed by averaging the number of medium level extrema over a 10x10 window.

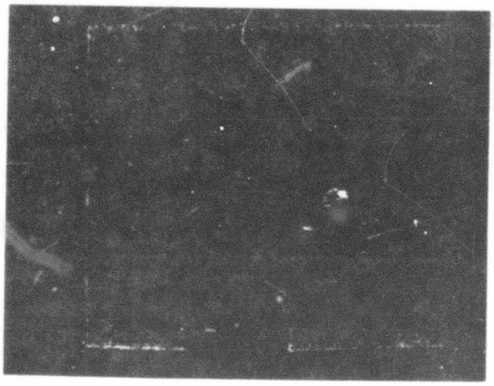


Fig. 6. Location of all points in Fig. 1 having a grey level-texture-edge combination occuring less than 15 times over the entire frame.

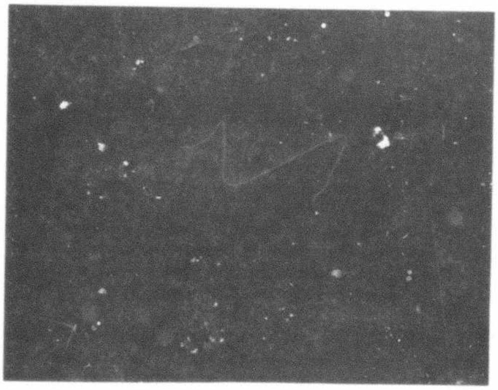


Fig. 7. Location of all points in Fig. 2 having a grey level-texture-edge combination occuring less than 15 times over the entire frame.

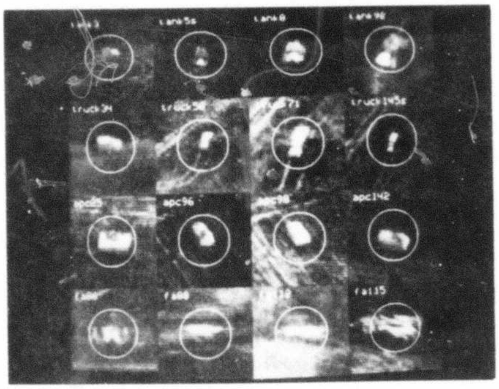


Fig. 8. Composite of 16 original potential targets. Each subpicture is 128x128 pixels. The top row consists of tanks, the second row consists of trucks, the third row consists of APC's and the fourth row shows false alarms.

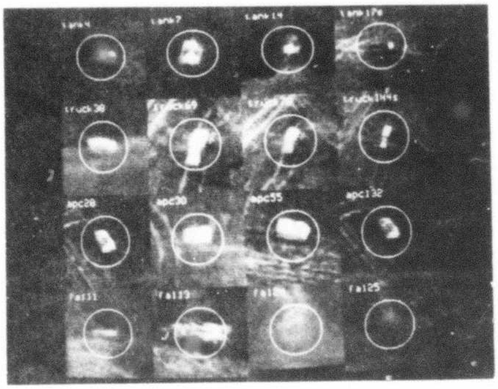


Fig. 9. Composite of an additional 16 original potential targets. The order by rows is again tanks, trucks, APCs and, false alarms.

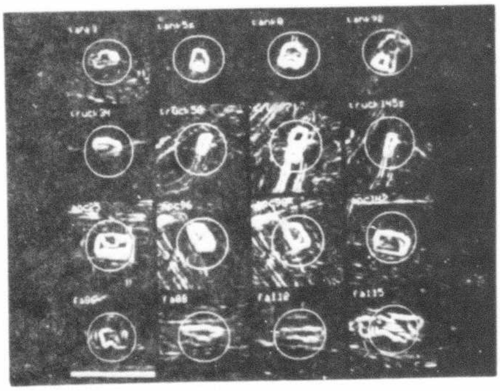


Fig. 10. Composite edge feature pictures from Fig. 8 using a smoothed gradient measure over a 7x7 window.

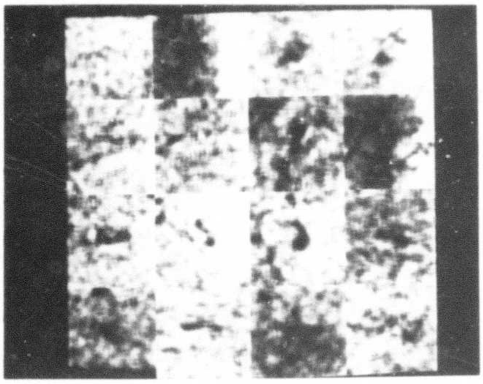


Fig. 11. Composite texture feature pictures from Fig. 8 formed by averaging the medium level local extrema over a 10x10 window.

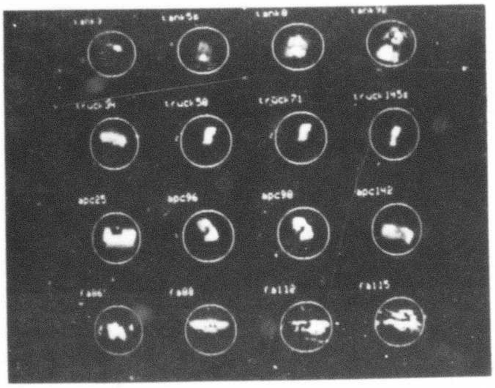


Fig. 12. Composite of segmented pictures derived from the grey level, edge, and texture features. The detected target points are shown in original grey levels; the background is shown as black. The original is Fig. 8.

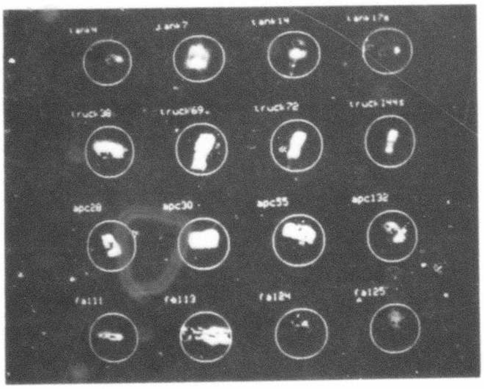


Fig. 13. Sixteen additional segmented pictures using the originals in Fig. 9.

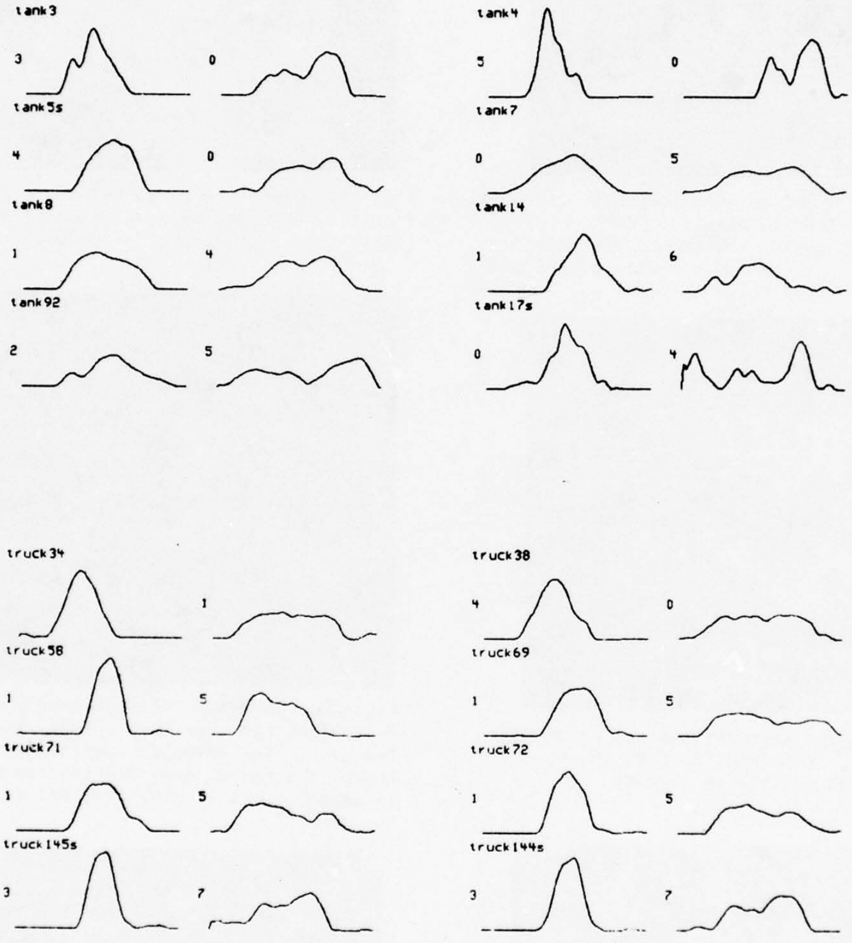


Fig. 14. Narrow (left) and wide (right) projections for all 32 segmented potential targets shown in Figs. 12 and 13. The digit to the left of each projection represents the direction of the projection (see text). This figure is continued on the following page.

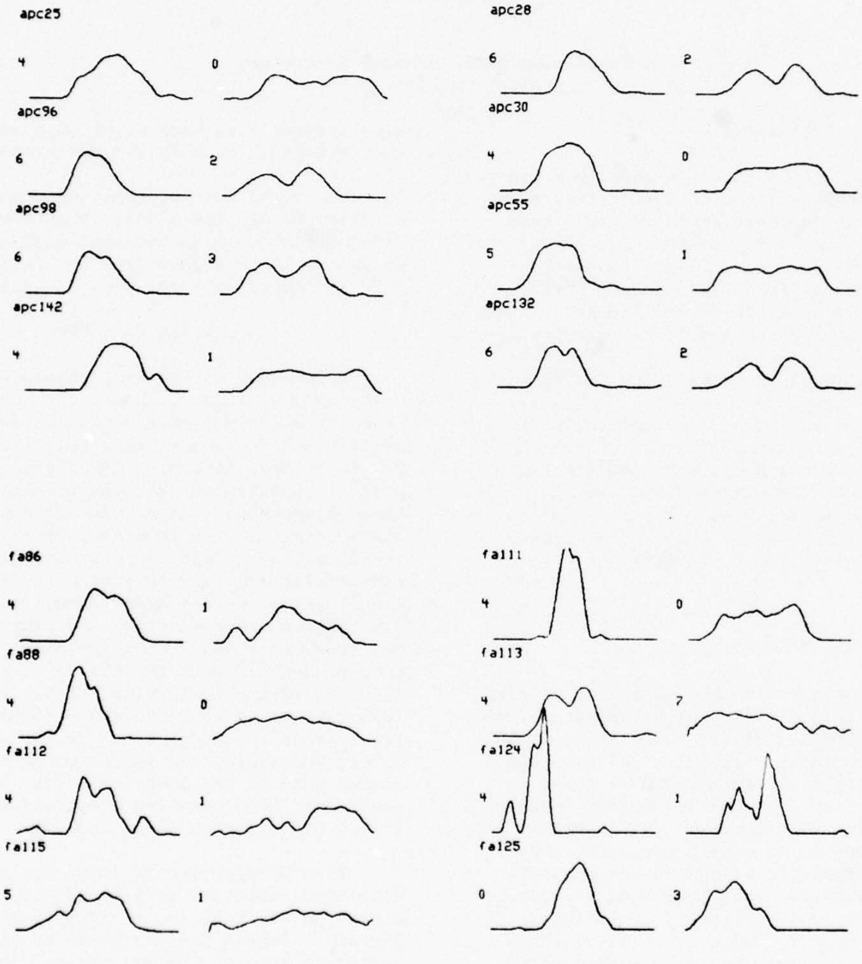


Fig. 14. (continued). Narrow and wide projection for the segmented potential targets.

ARTIFICIAL INTELLIGENCE CONCEPTS APPLIED TO NAVIGATION USING PASSIVELY SENSED IMAGES

Oscar Firschein
James J. Pearson

Lockheed Palo Alto Research Laboratory
Palo Alto, California 94304

ABSTRACT

A recently initiated study is described, investigating an aircraft navigation system that uses velocity and altitude measurements derived from passively sensed images of the terrain. Dead reckoning with periodic position fixing is the basic navigation approach. Three techniques, previously studied separately by Lockheed and Stanford University, will be combined and applied to the navigation problem. These techniques are (1) Image velocity correlation measurements for determination of the velocity to altitude ratio (V/H), (2) stereo vision distance measurements for altitude determination, and (3) curve segment representational matching for waypoint location. This combination of techniques could have future use in autonomous vehicles such as cruise missiles or remotely piloted vehicles (RPV) as a primary or secondary passive mode of navigation.

INTRODUCTION

A joint industry/university team, consisting of the Lockheed Palo Alto Research Laboratory and the nearby Stanford Artificial Intelligence Laboratory, will use their respective velocity and altitude measuring techniques based on passive imagery to delineate a navigation system based on the dead reckoning (DR) concept. An important part of the system is an expert navigator subsystem that combines the redundant measurements and initiates recovery procedures when the sensor data is missing or contradictory. Positional corrections are made by means of occasional map matches against reference imagery, supplemented by following distinctive features such as rivers or highways. The system description and operational aspects are given later in the paper.

The study is based on the following three image-based measurement systems:

1 - The Image Velocity Sensor (IVS) developed at the Lockheed Palo Alto Research Laboratory (PARL) [Ref. 1]. This approach uses a fast mechanization of phase correlation to obtain the velocity to altitude ratio (V/H) at video frame rate speeds.

2 - The stereo range and height determination system developed by the Stanford Artificial Intelligence Laboratory (AIL) [Ref. 2]. This approach uses two sensors mounted a fixed distance apart to view a scene simultaneously, or a single sensor that senses a scene at two difference times.

3 - The lineal feature tracking concept developed at Stanford AIL for a factory automation application [Ref. 3]. This approach will be extended to provide a curve segment representation as the reference data base for position fixing.

GENERAL BACKGROUND

Previously tested image-based navigation systems fall into two basic categories: (a) The tracker, which provides a continuous flow of position-estimate data and (b) the intermittent fix-taker Dead Reckoning (DR), which provides periodic position-update measurements. Both of these approaches have particular advantages and disadvantages. The less expensive tracker concept requires the storage of reference data covering essentially all terrain area along the planned flight path. On the other hand, the intermittent fix-taker concept requires the storage of much less reference data - but the operation of the concept depends upon the use of an expensive Inertial Navigation System (INS). This study investigates an approach which combines the best features of the tracker and the intermittent fix-taker, and avoids the shortcomings of continuous fixing such as the amount of data collected and prestored, and excessive computation for many fixing aids, particularly imagery.

If a DR approach to image navigation were developed, detailed data would probably be prestored on the vehicle for less than 20% of the terrain. Sensed imagery would be used continuously to support the DR process, but no stored reference imagery would be required until fix time. Specifically, ground speed and absolute altitude can support highly accurate DR without requiring pre-stored imagery.

The DR module continuously solves for the wind while it is receiving image derived ground speed. When the ground speed imagery is unreliable, the system uses the best known wind to solve for the ground speed. This calculated ground speed is then used until reliable image data is again available. At fix time, the DR position is used as an acceptance criterion for the fix. If the DR and fix positions differ by some unacceptable amount, a new fix is obtained. This prevents gross fix errors from being accepted by the navigation system. In an image-based system, this technique compensates for the fact that many sections of the surface terrain are mutually ambiguous.

Most automated navigation systems to date have adopted either a continuous fixing arpproach or a perfect DR approach. Loran, Omega and celestial tracking systems have been fully automated, but they do not use DR as a backup. If the fixing data is lost, the system freezes the last known data until fixing information is again available. During the reacquisition phase, false lock-on can occur. For example, in a Loran system a ground-wave-skyway mismatch can produce a 10 nautical mile error; accurate DR is essential if the false Loran lock-on is to be prevented. Inertial and Doppler systems attempt to provide automation through perfect DR. These systems, however, degrade as a function of time because of heading errors, data tracking servo errors, and data approximation assumptions during sensor non-lock-on periods.

These problems can be avoided by mimicking human approaches to navigation. This has not been done in the past because the software required to match the intelligence of a human navigator has been beyond the state-of-the-art of conventional software technology. However, during the past decade artificial intelligence researchers have developed software mechanisms for problem solving and deductive inference that can emulate many of the judgment mechanisms utilized by the human navigators.

This research effort will develop an artificial intelligence-based navigation system that blends DR and fixing navigation methods. The primary sensor will be passive imagery for both the fixing and DR mode. Passive imagery is preferred to provide covertness. If radar is needed for all-weather capability, or imagery is to be sent back to a photo interpreter, short-burst, single image transmissions would obtain a single terrain image. The automated DR system will minimize transmission requirements for the radar/communications system by telling the sensor when it is in a fix region. This automated DR system will allow more efficient Image Bandwidth Reduction since only those locations deemed important beforehand need to be encoded and sent back to a photo interpreter. This efficiency can lead to a much higher level of communications jam resistance for autonomous reconnaissance missions.

DESCRIPTION OF THE SYSTEM

The main components of the navigation system are shown in Fig. 1. Both image-based and conventional sensors feed information concerning altitude, air speed, heading, wind speed, and wind angle to the Measurement Management and Navigation Computation subsystem. In addition, image-based position checks based on image correlation and on lineal sketch matching are made periodically.

The key image-based measurement devices from the point of view of dead reckoning are the image velocity sensor that computes velocity to altitude ratio (V/H), and the stereo image analysis sensor that computes altitude (H). These two sensors will be the initial focus of the study. If the ground velocity of the vehicle could be reliably computed using these image-based instruments, there would be no need for other velocity-deriving sensors. However, because there may be times when the image based velocity sensors are inoperative due to the terrain characteristics, and we must, as in a Doppler navigation system, use the conventional air speed and heading, combined with an estimate of the wind velocity to perform the dead reckoning computations. Wind velocity can be determined from imagery by noting the "drift", or by performing "pressure pattern analysis"*.

As far as positional checking is concerned, a necessity due to the long time of flight, conventional phase or area correlation will be used, and no special study will be devoted to this topic. "Lineal sketch matching" a Stanford concept, now in the speculative stage, would permit representations of curved lines extracted from a sensed image to be compared against pre-stored representations of the overall region of operation. Highway or river following will also be considered as a possible position checking mode.

Combining these redundant measurements will be done using recently developed concepts in "analytic redundancy" [Ref. 4] combined with the heuristic AI concepts, such as used in error recovery in robots [Ref. 5].

The operational characteristics of the various subsystems are given in Table 1. It should be kept in mind that the data rates indicated are rough estimates prior to analysis. An important aspect of many sensor outputs, as indicated, is that a confidence measure is provided with each measurement. This is necessary so that the navigation management subsystem can make decisions based on these confidence measures.

MIMICKING THE HUMAN NAVIGATOR

The human navigator can monitor flight progress of a mission by using a flight plan graph, (Fig. 2). Basically the flight plan graph consists of a line representing the flight plan time from departure to destination or turning point and roughly paralleling the true course of a flight. Predicted times to various points along

* On the basis of pressure measurements made at two points and on the known Coriolis effect at a given latitude, it is possible to compute the "geostrophic wind", the wind assumed to blow parallel to the isobars. "Pressure pattern flying" assumes that the geostrophic wind approximates the true wind for latitudes above 20 degree N. and below 20 degrees S. The drift perpendicular to the air path due to the wind is proportional to the difference between the absolute altitude (as measured by stero imagery) and pressure altitude (as measured by barometric altitude sensors) at two points.

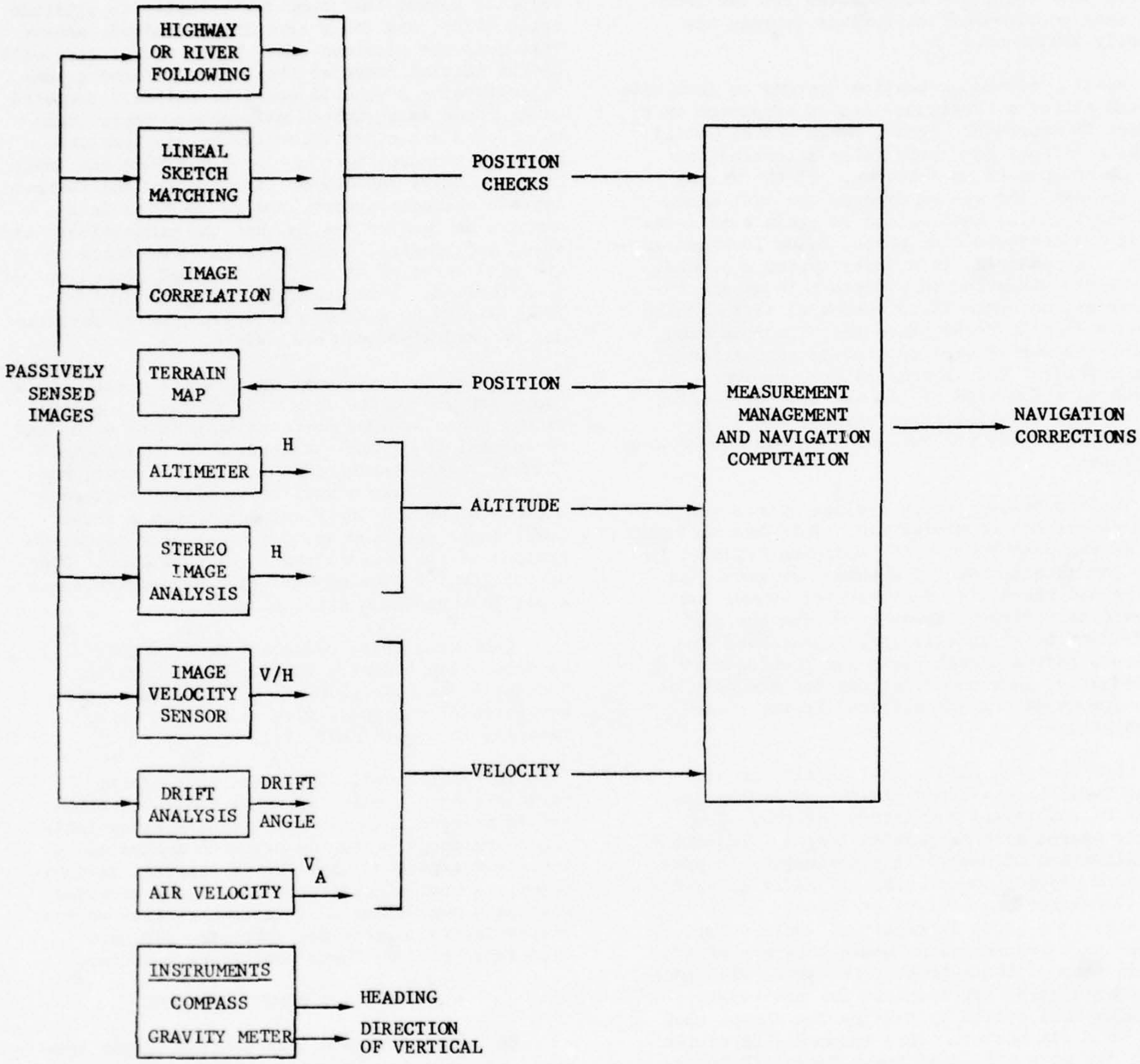


Fig. 1 Overall Navigation System

the true course, together with departure and destination times, are plotted on this time scale. Thus, the flight plan graph represents a visual time line comparable to the predicted track. Using the time line, the predicted estimated time of arrival to any point on the predicted track can be determined. Comparison with a fix, checkpoint, or obstacle gives the aviator or observer an indication of whether he is ahead or behind his flight plan, and whether he is on course. (Good and poor visual checkpoints for the human navigator are tabulated in Table 2. However, it will be noted that some of them would be difficult to detect automatically).

In the proposed system, the data navigation manager must construct and follow a representation that is the equivalent of the flight plan graph. This representation is constructed from the sketch-map information for the overall region, based on the desired flight path. The graph may have to be revised several times during flight by the navigation manager as the flight plan is changed. Besides landmarks, the graph should also indicate when strong features such as rivers or roads are likely to be encountered that might serve for feature following. This graph also is used to note when the vehicle will be in the region of a stored reference map for use by the map matcher. Thus, the flight plan graph serves the navigation manager as a master scheduling, landmark noting, and trouble diagnosis aid.

TABLE 1

Operational Characteristics of the Subsystem

Subsystem	Operational Characteristics	Output
Image-based velocity sensors	Operational at all times. IVS outputs 10-30/sec.; stereo altitude 1-10/sec.	V/H and confidence factor H and confidence factor
Conventional air speed and heading	Operational at all times. Sampled 10/sec.	Air speed and heading
Position fixing using map-matching	Operational only when vehicle is in region of a stored reference map.	Distance displacement error and confidence factor
Sketchmap matching		
a. general location	Operational at all times. Results every minute or so.	General indication of path validity, plus confidence factor
b. feature following	Operational when requested by Navigation Manager, or when strong feature is found. Results 5-10 secs.	Navigation corrections and confidence factor
Pressure pattern analysis	Estimates available every 1-5 min.	Wind estimates and confidence factor

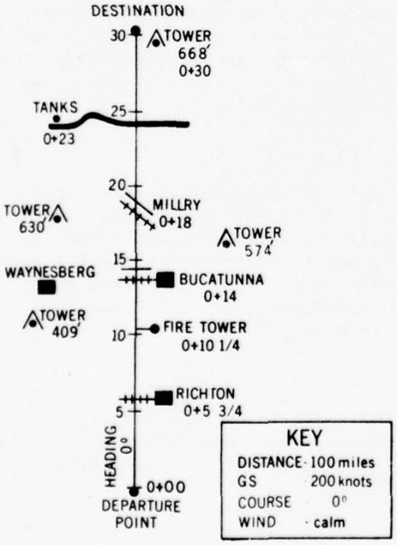


Fig. 2 Flight Plan Graph

QUESTIONS TO BE ANSWERED

The following questions must be answered to determine the feasibility of the image-based navigation concept:

Image Velocity Sensor

- What accuracy in V/H ratio can be expected, given the perturbations of the aircraft in pitch, roll, and yaw?
- Can smoothing techniques based on knowledge of the vehicle dynamics be used to improve the computed V/H ratio?
- How sensitive is the accuracy of the V/H determination to the nature of the sensed image?
- Can good measures of dependability of V/H be derived for use by the navigation management system?

Stereo Altitude System

- How accurate can the altitude determination system be using two vision sensors?
- Is a single sensor system (multiple looks separated in time) practical?
- How sensitive is the altitude computation to the nature of the imagery?
- Is a simple confidence measure available from the stereo altitude device for use by the navigation management system?

Line Sketch Analysis

- What techniques for extraction of lineal features from imagery should be used?
- What non-image representation should be used to store and compare these features?
- What positional accuracy can be expected from feature matching?

TABLE 2

Good and Poor Visual Checkpoints for the Human Navigator

GOOD CHECKPOINTS	POOR CHECKPOINTS
MOUNTAINOUS AREAS	
Prominent peaks, cuts and passes, gorges. General profile of ranges, transmission lines, railroads, large bridges over gorges, highways, lookout stations. Tunnel openings and mines. Clearings and grass valleys.	Smaller peaks and ridges, similar in size and shape.
COASTAL AREAS	
Coastline with unusual features. Lighthouses, marker buoys, towns with cities, structures.	General rolling coastline with no distinguishing points.
SEASONAL CHANGES	
Unusually shaped wooded areas in winter. Dry river beds if they contrast with surrounding terrain. Dry lakes.	Open country and frozen lakes in winter unless in forested areas. Small lakes and rivers in arid sections of country - in summer - when they may dry up. Lakes (small) in wet seasons in lake areas, where ponds may form by surface waters.
HEAVILY POPULATED AREAS	
Large cities with definite shape. Small cities with some outstanding checkpoint; river, lake, structure, easy to identify from others. Prominent structures, speedways, railroad yards, underpasses, rivers and lakes. Race tracks and stadia, grain elevators, etc.	Small cities and towns, close together with no definite shape on chart. Small cities or towns with no outstanding checkpoints to identify them from others. Regular highways and roads, single railroads, transmission lines.
OPEN AREAS; FARM COUNTRY	
Any city, town, or village with identifying structures or prominent terrain features adjacent. Prominent paved highways, large railroads, prominent structures, race tracks, fairgrounds, factories, bridges, and underpasses. Lakes, rivers, general contour of terrain; coastlines, mountains, and ridges where they are distinctive.	Farms, small villages rather close together, and with no distinguishing characteristics. Single railroads, transmission lines and roads through farming country. Small lakes and streams in sections of country where such are prevalent, ordinary hills in rolling terrain.
FORESTED AREAS	
Transmission lines and railroad right-of-ways. Roads and highways, cities, towns and villages, forest lookout towers, farms. Rivers, lakes, marked terrain features, ridges, mountains, clearings, open valleys.	Trails and small roads without cleared right-of-ways. Extended forest areas with few breaks or outstanding characteristics of terrain.

- How serious are perspective effects caused by roll and pitch on the matching procedure?

- For what type of terrain is the sketch approach infeasible?

Navigation Manager

- How can formal methods for combining redundant measurements be applied?

- For what modes of operation should heuristic AI approaches be used?

- What recovery techniques used by human navigator can be automated?

Wind Analysis

- What formal and informal methods of computing wind velocity can be automated, given the imagery and the standard aircraft instrumentation available?

- What strategies for estimating and extrapolating wind velocity can be used when image based measurements are not available?

APPROACH

The study and experimental verification of the use of imagery for navigation purposes includes work in three areas: velocity sensing, altitude sensing, and mapping, and this will lead to an analysis of the mechanization of this dead reckoning approach for future airborne systems. The effort will have six phases, performed over a time period of 27 months. The six tasks are as follows: (I) Image acquisition, (II) Image velocity sensor (IVS) experiments, (III) Stereo vision experiments, (IV) Lineal mapping studies, (V) Artificial intelligence (AI) approaches to redundancy management and integration of additional sensor inputs, and (VI) Mechanization analysis and evaluation.

Tasks III and IV will be performed by Stanford AIL, while the other tasks will be performed by the Signal Processing Laboratory of Lockheed PARL. A brief description of each task is as follows:

Task I - Image Acquisition. Suitable high resolution aerial photography will be obtained and digitized. Maximum use will be made of the imagery available within Lockheed and Stanford, and that available through ARPANET from other sources. The imagery will be used for the experiment in the following tasks. *(See Appendix A)*.

Task II - Image Velocity Sensor (IVS) Experiments. Experiments will be conducted using the IVS system developed by Lockheed. The experiments will test the sensitivity of IVS to various perturbations in the attitude of the sensor vehicle, to time errors in the sensing and processing of imagery, and to altitude errors in uneven terrain and in cloud-masked images.

Task III - Stereo Vision Experiments. Experiments will be carried out using Stanford AIL stereo vision system to measure vehicle altitude. The experiments will test the sensitivity of the altitude measurements to terrain irregularities, percentage of ground surface visible, timing (distance) error in the sensing of the two images, image variation, and vehicle attitude perturbations.

Task IV - Lineal Mapping Studies. Stanford AIL will study the feasibility of using a lineal mapping system to derive a map and its compact representation and compare the representation to the representation of a known map of the area. The feasibility of using this method as a waypoint location technique for navigation will also be considered.

Task V - AI Approaches to Redundancy Management. This task will develop the conceptual design for a data management system to perform the navigation DR task. This design will consider the use of other sensor-derived information, such as pressure pattern analysis and highway following. Where relevant, the management system will use AI techniques to make the navigation decisions based on the noisy sensory-derived measurements.

Task VI - Mechanization Analysis and Evaluation. The navigation system will be examined from an implementation point of view, including estimates of size, weight, power, and cost. These figures will be used to evaluate the practicality (operational and maintenance) of implementing such a system for autonomous aircraft. An estimate of the midcourse and terminal accuracy of the navigation system will be prepared.

Two laboratory demonstrations will be prepared. The first one will cover the work of the first year in Task II and Task III, and the second demonstration at the end of the second year will cover the remainder of the effort. Because of computer system incompatibilities, the demonstrations will not show the total integrated system, rather, the stereo and lineal sketch map aspects will be demonstrated at the Stanford AIL facilities, and the IVS and management aspects of the system will be demonstrated at the Lockheed PARL facilities. Necessary data from the Stanford experiments will be shared using magnetic tape; thus, derived altitude information will be pre-stored for the Lockheed experiments.

SUMMARY

A study using passively sensed images as the basis of an aircraft navigation system has been described. A crucial question to be answered is whether image-based measurements are usable if the image sensors are not inertially stabilized. In order to clarify and summarize the key elements of this study, similarities and differences between the present study and existing DARPA terminal homing studies are given in Table 3.

REFERENCES

1. J. J. Pearson, D. C. Hines, B. S. Golosman, and C. D. Kuglin, "Video-Rate Image Correlation Processor", SPIE, Vol. 119, Application of Digital Image Processing (IOCC 1977).

2. D. Gennery and T. Binford, Image Understanding Workshop, "Stereo Vision System", October 1977.

3. R. C. Bolles, "Verification Vision", Artificial Intelligence Laboratory, Memo AIM-295, Dec. 1976.

4. J. J. Deyst and A. L. Hopkins, Jr., "Highly Survivable Integrated Avionics", Astronautics and Aeronautics, September 1978.

5. S. Srinivas, "Error Recovery in Robots Through Failure Reason Analysis", National Computer Conference, 1978.

APPENDIX A
IMAGERY REQUIRED FOR EXPERIMENTS

In selecting imagery, one is faced with the usual dilema of running experiments under controlled conditions using "artificial" imagery, versus using realistic but uncontrolled imagery. Our initial thoughts concerning the imagery

TABLE 3

Relation of Present Study
to DARPA Terminal Homing Studies

	Topic	Lockheed/Stanford Study	DARPA Terminal Homing Studies
Similarities	Applications Area	The general applications area is the guidance of a vehicle such as a cruise missile, characterized by long flight time, low altitude, relatively slow speed, and a range of 500-1500 miles	
	Positional Fixing Technique	Map-matching is used for positional fixing.	
Differences	Navigation vs. Target-Looking Terminal Homing	Problem focus is aerial navigation using dead reckoning	Problem focus is target-looking terminal homing
	Near-Term vs. Long-Term Concepts	Several techniques are speculative and unproven, and are being investigated for possible future systems.	Proven, near-term techniques are being used.
	Guidance System	The dead reckoning navigation system is a principal focus of the study.	The inertial guidance system is not a principal focus of the study.
	Redundant Sensors	Massive and distinct sensor redundancy is managed by an AI or oriented management system.	Non-redundant use of sensor.
	Feature Extraction	Use of feature identification (e.g., highways and rivers) for both following and matching.	Where feature extraction is used, features are used for matching.
	Vehicle Path	Flexible, non-programmed path.	Pre-programmed path.

required are as follows:

Image Velocity Sensor. The IVS requires a sequence of images that overlap by about 70% so that accurate correlations can be made. We can simulate this using a 512 x 512 pixel image and selecting overlapping 128 x 128 windows from the image. To add realism, we can add random noise as we go from window to window.

If the centers of the windows chosen lie on a straight line, then we are simulating level flight with zero pitch, yaw, and roll. By perturbing the window centers in one direction, we can simulate vehicle roll; by perturbing in the other direction, we can simulate vehicle pitch. We can then examine the positional errors obtained by integrating the instantaneous velocity as well as the velocity values obtained by smoothing the instantaneous velocity measurements. (Integrated velocity gives vehicle position; smoothed velocity provides an independent estimate of wind velocity).

After these controlled experiments are performed, we can use a sequence of TV frame rate images taken from a stabilized platform as a source of realistic (but uncontrolled) imagery.

Stereo Altitude. The Stanford AIL experiments will determine whether altitude measurements can be made using unstabilized sensors, when the pitch, roll, and yaw of the vehicle are not known. A pair of onboard sensors is required in the operational system to compute altitude by using a camera model to develop a "ground plane". Thus, for this experiment, we need imagery from an unstabilized platform traveling at a constant altitude and consisting of stereo pairs taken by a stereo camera.

To examine the effects of sensor separation, we can use the TV frame rate imagery used in the IVS experiments. Sensor separation can be simulated by skipping frames.

SESSION III

TECHNIQUES II

LINEAR FEATURE EXTRACTION

by

Ramakant Nevatia
K. Ramesh Babu

Image Processing Institute and Computer Science Department
University of Southern California, Los Angeles, California 90007

INTRODUCTION

In previous work [1], we have described initial attempts at locating objects of interest in aerial images. The effectiveness of this system, and of similar systems developed elsewhere, seems to be strongly limited by the power of low level processing programs. In this paper, we describe a set of procedures for extraction of linear features, useful for detection of roads and runways for example, and believe that it will help in substantial improvement in the overall system performance.

In spite of the large amount of previous research in this area, no algorithms suitable for complex imagery are apparent. In particular we found the widely used Hueckel operator to be deficient for images with fine detail and texture. The described algorithms seem to achieve better performance on a variety of images, and are already being used by Hughes Research Laboratoris on an independent ARPA program and being considered for use by Tom Binford's group at Stanford.

The process of line finding consists of determining edge magnitude and direction by convolution of an image with a number of edge masks, of thinning and thresholding of these edge magnitudes, of the linking of the edge elements based on proximity and orientation, and finally of approximation of the linked elements by piecewise linear segments. Some objects of interest, e.g. roads and runways, are characterized by being bounded by nearly parallel line segments of opposing contrast, to be known as anti-parallel segments. Our algorithms are largely local in nature and can be applied to large images without difficulties of storage (but, of course, requiring proportionately larger computing time), and hardware implementation should be feasible. These algorithms are presented here as pragmatic solutions to the low level problems of image understanding with little discussion of their optimality or novelty.

EDGE DETECTION

Edge detection is done by convolving a given image with masks corresponding to ideal step edges in a selected number of directions. The magnitude of the convolved output and the direction of the mask giving the highest output at each pixel are recorded as edge data. (The edge data are two files, one containing the magnitude and the other, a coded direction). We have found 5 x 5 masks in six directions as shown in Fig. 1 to be suitable for most images of interest. The choice of mask sizes needs to be investigated further. In general, the small masks are more sensitive to noise whereas the larger masks cannot resolve fine detail and may have difficulties if the texture elements are of similar size. We have chosen not to use the techniques of adaptive mask size selection by comparing the outputs of a large number of masks of varying size as suggested by Rosenfeld and Thurston [2] and by Marr [3], due to unacceptable computational requirements for large images. The criteria for choosing from among the many sizes are also unclear in presence of texture. However, use of more than one mask size may be necessary for certain applications.

THINNING AND THRESHOLDING

The presence of an edge at a pixel is decided by comparing the edge data with some of the 8 neighboring pixels. An edge element is said to be present at a pixel if:

1. the output edge magnitude at the pixel is larger than the edge magnitudes of its two neighbours in a direction normal to the direction of this edge. (The normal to a 30 degree edge is approximated by the diagonals on a 3 x 3 grid);

2. the edge directions of the two neighboring pixels are within one unit (30 degrees) of that of the central pixel; and

3. the edge magnitude of the central pixel exceeds a fixed threshold.

Further, if the conditions 1 and 2 above are satisfied, the two neighboring pixels are disqualified from being candidates for edges. This algorithm produces results independent of the order in which the pixels are examined.

A more judicious decision could be based on examining the shape of the profile of convolution output, e.g., an ideal step edge should produce a triangle-shaped output. Such techniques have been used by Herskovitts and Binford [3] and by Marr [2]. Our experiments with requiring the neighboring pixels to have edge magnitudes that are at least a certain fraction of the central pixel magnitude resulted in poor performance perhaps due to variations caused by fine texture in the test images. More complex decision strategies hold promise for improved performance.

LINKING

A boundary in a digital plane is a collection of points where each point is connected to two of its 8-neighbors. (Except for edge points and where "forks" exist). One approach to connect up such points, therefore, is to determine the two neighbors for each edge point. The two neighbors can be further distinguished as a predecessor and a successor. The boundary is then a threading through these edge points using this information.

The primary aspect of the linking process is the determination of a predecessor and a successor, if any, at each edge point. We produce two matrices - p and s - of the same physical dimensions as the image. (We have stored them as p and s files on the disk). Our criteria for connecting two edge points is that they be neighbors, in the 8-neighbor sense, and that they have edge directions differing by not more than a certain value, currently set at 30 degrees for masks described previously. Due to the nature of thinning, only three locations are potential candidates for predecessor or successor elements as shown in Figs. 2(a) and (b) for edges of 0 and 30 degree directions, respectively. The determination of successor (predecessor) pixels is elaborate due to the several cases that are possible at each pixel:

1. Only one element is an acceptable successor. In this case the successor (predecessor) is recorded in the s(p) file as an integer between 0 and 7 corresponding to its location.

2. Two candidates are acceptable successors. If they are not 4-neighbors, a fork is present as shown in Fig. 3(a). If they are 4-neighbors, a fork exists only if their directions differ by more than 2 units (60 degrees), as in Fig. 3(b). Otherwise no fork exists and the nearer of the two (using Euclidean distance), forms the successor (predecessor), as shown in Fig. 3(c). These rules are for smooth continuation of lines and were derived by complete enumeration of such configurations. In case of a fork the stronger of the two candidates in edge magnitude forms the main stream. The fact that a fork exists is noted in the s(p) file. This information is sufficient to trace both streams of a fork by examining the p and s files simultaneously.

3. Three candidates are acceptable successors. Fig. 4 shows all possible such configurations for a vertical edge (no three successor configurations occur for 30 degree edges). In these cases, a fork exists. The main stream is formed by the nearer of the two edges having the same direction, and the other candidate with different direction forms the other branch.

Note that this representation of the linked edge elements is in contrast to explicit lists of elements forming a connected segment. For large images, not entirely resident in core, it is more convenient to form predecessor and successor matrices as the processing requires only a sequential scan of the image file. Further, certain proximity computations can be more easily performed using the predecessor and successor files.

We now describe briefly how we can make use of the p and s matrices to produce a one-time traversing of all the curves in the picture. Such a traversing is necessary both to obtain a display on a suitable device and in fitting linear segments to the curves as described later. The general scheme is a TV raster scan which looks for the condition for starting a traversal:

```
var rscan: 1..noofrows;
    cscan: 1..noofcolumns;
for rscan:= 1 step 1 until noofrows do
begin
  for cscan := 1 step 1 until noofcolumns do
  begin
    if start(rscan,cscan) then
    repeat
      visit this pixel;
      compute next pixel;
    until cannot proceed;
  end;
end;
```

The above algorithm is applied to the p and s files in three passes, with a different predicate "start" to decide if traversing should start at a pixel. In the first case, a traversing starts when a pixel does not have a predecessor but has a successor. The second pass examines if the predecessor was a fork point, and thus picks up the secondary branches. The final pass starts traversing at those pixels that have not been "visited" previously and picks out circular segments. Information about previous visits is stored in a temporary binary file. During any pass, we "cannot proceed" if we come to a pixel that has already been visited.

FITTING PIECEWISE LINEAR SEGMENTS

If we are looking for straight edges in the picture, we need to fit piecewise linear segments to the (digital) curves that we obtain after linking, as described above. We have used a version of the iterative end-point fits algorithm of Duda and Hart [4]. A point on a digital curve is a corner if it is the most-removed from the endpoints. The first corner in a curve thus produces two segments both of which can contain more corners and so a recursive application of the same procedure is appropriate:

```
type point = record
               r:  rowcoordinate;
               c:  columncoordinate;
             end;
```

```
procedure cornersinwindow(pl, p2: point);
var p: point;
begin
  p := pl;
  repeat
    p := next(p);
    if p is a corner then
    begin
      mark p as a corner;
      cornersinwindow(pl,p);
      cornersinwindow(p,p2);
    end;
  until p = p2;
end;
```

A straightforward application of such a recursive procedure can be inefficient. On an average, it takes $o(n^2)$ time to process a curve which is n points long. Hence, a variation which embodies the above mentioned qualities but is superior is employed. Instead of considering the entire curve and then applying the above procedure we apply it on a smaller portion of the entire curve, say m points long. For the next part of processing, the curve begins at the farthest corner found, and ends m points later and so on, until the end of the original curve is reached. To avoid the possibility of the algorithm missing some corners because the end point of an m-long portion was at or around a genuine corner, we consider 2m-long chains in case no corners are found, then chains 3m-long...and so on, until either we find a corner or come to the end. A typical value for m is 32 elements. We believe this algorithm to substantially faster on the average, but have not yet performed a detailed analysis or comparison.

On output a segment is described by a unique id, its predecessor or successor segments along the flow of the curve, coordinates of the end points, length and direction.

SOME RESULTS

Results of processing an airport image at various stages of processing are shown in Figure 5. The computation times for various stages of processing are as follows (for a 128 x 128 image, on a PDP-10, KL-10 processor):

Convolution with edge masks	17 secs.
Thinning and Thresholding	2.3 secs.
Linking (p and s files)	2.2 secs.
Segment tracing and Linear approximations (maximum error-2 pixels)	4.8 secs.

All computation times, except for linear segment fitting, scale linearly with the number of points to be processed. Also, except for the linear segment approximation, the storage requirements are limited to only a few lines of an image at a time.

FINDING ANTI-PARALLEL PAIRS

The first step is to sort the segments by their orientation. This sorting collects together segments that are potential matches to a given segment and hence avoid looking through the entire list in finding a match for the given segment. However, due to errors in the orientation of segments, sorting based on exact angles is unnecessary, and the segments with same angles correct to the nearest integer are grouped together.

In finding a pair of segments of antiparallel orientation we look for those whose angles are $(180\pm\alpha)^{\circ}$ apart, where α is a tolerance factor. Further, we require that the segments overlap and that they be within a certain distance of each other. These antiparallel pairs (apars) are then described as 2-dimensional generalized cones (see [6-8]) with an axis and a width and an additional attribute of relative brightness. A unique identifier is associated with each apar. Fig. 6 shows the axis of cones found from the segments shown in Fig. 5(f).

SELECTION AMONG ANTI-PARALLELS

Proper choice of apars that correspond to objects can be difficult and is like resolving figure-ground relationships (e.g. see Figs. 7(a) and (b). However, for many applications, such as for roads and runway detection, a choice of the closest pairs may suffice and may also be aided by knowledge of the desired objects being brighter or darker than the background. Also, the axes of the cones can be merged on the basis of collinearity to form larger cones. Work along these lines is currently in progress.

REFERENCES

1. R.Nevatia and K. Price, "Locating Structures in Aerial Images," ARPA Image Understanding Workshop, Palo Alto, Ca., October 1977, pp. 52-54.

2. A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," IEEE Transactions on Computers, Vol. 20, May 1971, pp. 562-569.

3. D. Marr, "Early Processing of Visual Information," MIT AI Memo No. 340, Dec. 1975.

4. A. Herskovitts, "On Boundary Detection," MIT AI Memo 183, 1970.

5. R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, Wiley, 1973.

6. R. Nevatia and T.O. Binford, "Description and Recognition of Curved Objects." Artificial Intelligence, Vol. 8, No. 1, February 1977, pp. 77-98.

7. R.A. Brooks, R. Greiner and T.O. Binford, "A Model Based Vision System," Proceedings of ARPA Image Understanding Workshop, May 1978, pp. 36-41.

8. D. Marr, "Analysis of Occluding Contour," MIT AI Memo 372, October 1976.

-100	-100	0	100	100
-100	-100	0	100	100
-100	-100	0	100	100
-100	-100	0	100	100
-100	-100	0	100	100

a) 0°

-100	32	100	100	100
-100	-78	92	100	100
-100	-100	0	100	100
-100	-100	-92	78	100
-100	-100	-100	-32	100

b) 30°

100	100	100	100	100
-32	78	100	100	100
-100	-92	0	92	100
-100	-100	-100	-78	32
-100	-100	-100	-100	-100

c) 60°

100	100	100	100	100
100	100	100	100	100
0	0	0	0	0
-100	-100	-100	-100	-100
-100	-100	-100	-100	-100

d) 90°

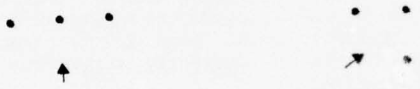
100	100	100	100	100
100	100	100	78	-32
100	92	0	-92	-100
32	-78	-100	-100	-100
-100	-100	-100	-100	-100

e) 120°

100	100	100	32	-100
100	100	92	-78	-100
100	100	0	-100	-100
100	78	-92	-100	-100
100	32	-100	-100	-100

f) 150°

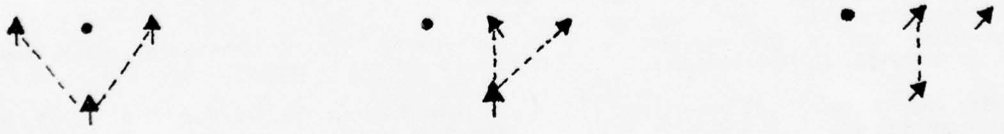
Fig. 1. Edge Masks in 6 Directions.



a) 0° Edge

b) 30° Edge

Fig. 2. Possible Successor Locations for Two Edges.

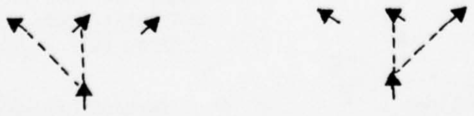


a) Non-neighboring Successors

b) Successors Directions Differ by 60°

c) Successors of Same Direction

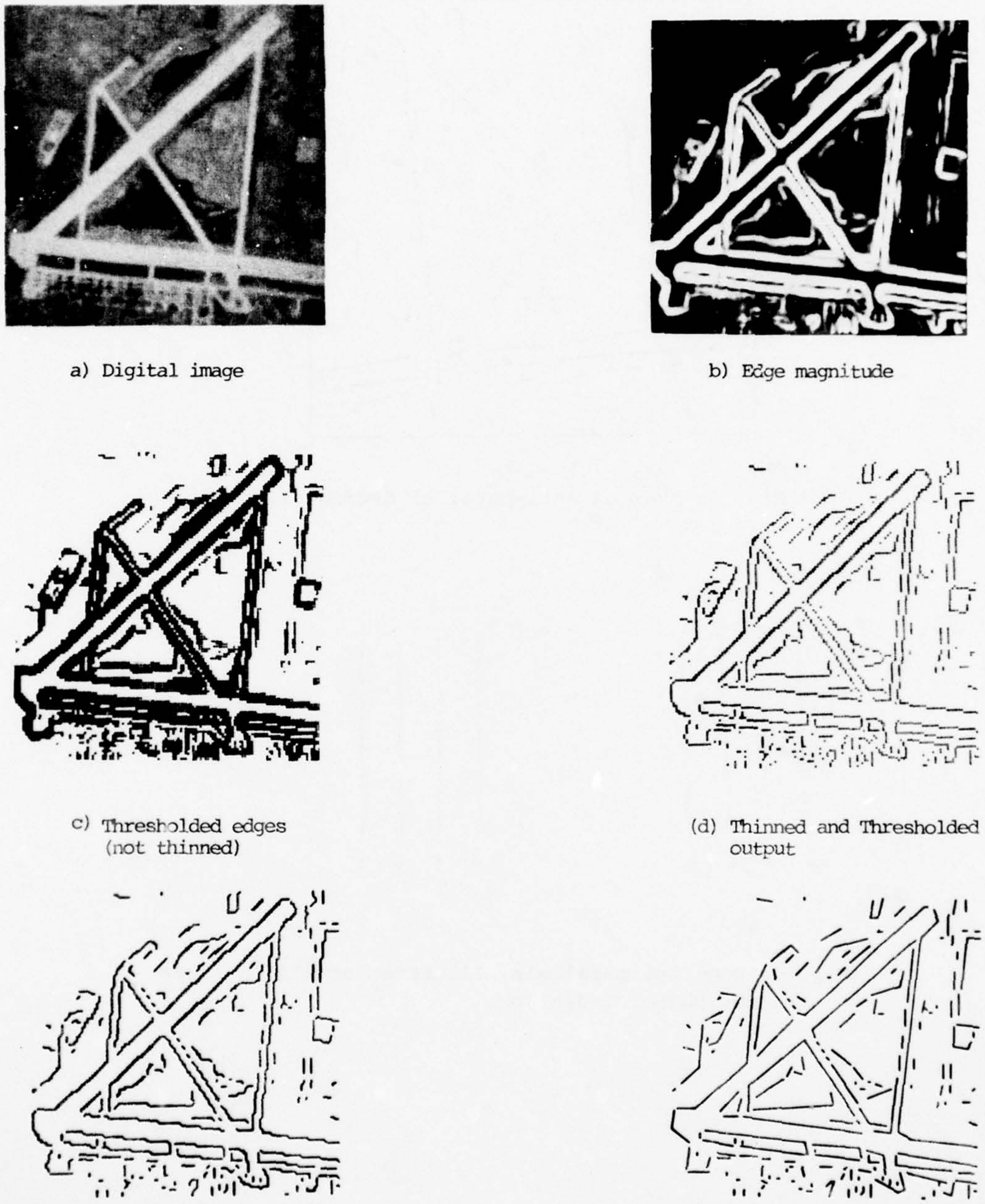
Fig. 3. Three Instances of Two Successors.



a)

b)

Fig. 4. All Instances of Three Successors.



a) Digital image

b) Edge magnitude

c) Thresholded edges (not thinned)

(d) Thinned and Thresholded output

(e) Linked segments

(f) Linear segment approximation

Fig. 5 An airport image and various stages of processing

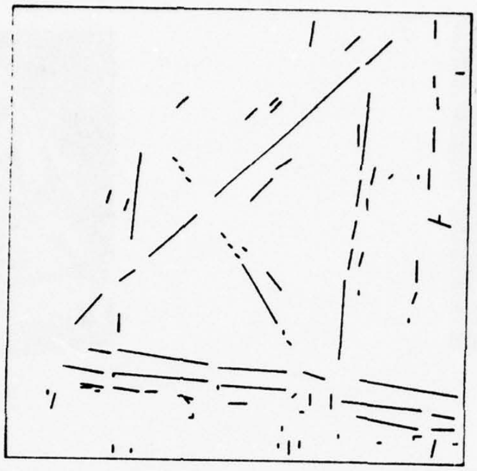


Fig. 6. Axes of Anti-parallel Segments.

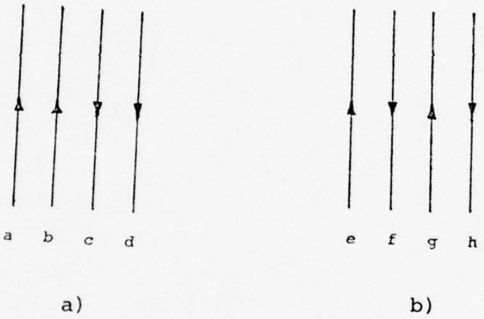


Fig. 7. Some Antiparallels, Illustrating Difficulties of Object Isolation.

SHAPE FROM TEXTURE:
A BRIEF OVERVIEW AND A NEW AGGREGATION TRANSFORM

John R. Kender

Department of Computer Science
Carnegie-Mellon University, Pittsburgh, Pa. 15213

Abstract

A new approach to obtaining shape information from textural information in static monocular images is outlined. Also presented is a new aggregation transform useful in the determination of vanishing points. Additionally, the transform has many properties that make it an appealling substitute for some other current image transforms. Examples are given of the application of the transform to both synthetic and natural images.

Introduction

One central task of image understanding is the recovery of three-dimensional scene information from the two-dimensional perspective transformation that is the image. The recovery of the missing dimension can be achieved by the use of multiple views: either extensive in time, as in the determination of structure from motion [Ullman, 1977], or extensive in space, as in deriving shape from binocular disparity [Gennery, 1977]. However, even a single image often contains powerful cues as to object definition and shape; for example, many properties of object surfaces can be derived from an understanding and exploitation of image intensities [Horn, 1977]. In so restricting the input, the task necessarily becomes a heuristic one, given the vast array of scenes that can generate identical images. (In the extreme, one is never certain if the "external scene" is not itself two-dimensional--that is, the image is a picture of a picture.) But such restrictions, if coupled with the demand that processing be relatively model-free, can provide basic theories, heuristics, and algorithms applicable to many other image tasks.

This paper begins with a very brief outline of one such low-level approach to deriving shape information from a static monocular view. This method, under development, is based on the analysis of texture gradients and the application of principles of projective geometry. It is hoped that just as the investigation of the reflectivity of surfaces and the analysis of the physics of the scene-image configuration enables shape to be derived from shading, shape can also be derived from the textural properties and the perspectivity of a scene.

The remainder of the paper is devoted to the presentation of a new image aggregation transform, in the style of the Hough transform. It is more efficient and "natural" than many existing Hough-like transforms, and is useful for determining the location of local or global vanishing points and lines. The determination of such points and lines is a necessary step in deriving surface gradient information from textural variations; they are intimate functions of the local or global gradient-space values [Mackworth, 1973].

Shape-related Aspects of Texture

"Texture" is an ill-defined term. However, in one respect, it can be considered an attribute of surfaces not unlike reflectance or color: its appearance is usually dependent on illumination and view angle. It is well known that blurred textures behave very much like gray scale tones; texture gradients are similarly likened to intensity gradients.

But there are also important, exploitable differences. Intensities are usually identified one-to-one with picture elements ("pixels"), and thus have no shape; further, because of the inverse square law, reflected luminance is independent of distance. It is difficult to discriminate intensity differences due to illumination variation, reflectance differences, or orientation. In contrast, consider textures, especially those made up of identifiable texture elements ("texels": in so-called "statistical" textures, this role is roughly filled by areas of local extrema). Texel definition can be rather insensitive to illumination variation; indeed, if individual texture components have negligible extent normal to the surface they define (the texture is reflectance variation: "paint"), even shadows may not obliterate the fundamental textural pattern. Further, texel density and orientation are highly correlated to orientation and distance; the inverse square law holds exactly. Texture gradients, like intensity gradients, can be smooth or abrupt; but, given the necessarily large area needed to define a texel they comprise a more than one-dimensional family, and are therefore potentially more discriminating at occlusions, and less sensitive to noise.

Intensity and texture are somewhat complementary, then, and often coexist within the same surface (e.g. the surface of a golf ball). Both can be, and need to be, exploited in order to understand the other.

At least three interrelated phenomena partake in the analysis of shape through textural information. First, there is surface integrity, pursued in the intensity domain by region-growing or -splitting approaches, and by classical shape-from-shading. The analogue for a textured object is based on the assumptions of local texel similarity. (Thus region-growing and -splitting implicitly define near-planar surfaces based on the similarity of very small texels: pixels). Secondly, there is surface orientation, derivable from the assumptions directly applied in shape-from-shading: local surface ("microplane") orientation uniqueness, and global surface continuity. Smooth changes in surface orientation will give rise to textural gradients, but not conversely, so heuristic rules are necessary. Lastly, there is surface location, in part derivable from those additional texture gradients occasioned by perspective deformation. Such gradients have no counterpart in the intensity domain. They can be analyzed by using the assumption of the uniqueness of viewer position, with respect to which surfaces have direction or distance.

Each of these three phenomena can be studied more or less in isolation by carefully selecting images that minimize the effect of the other two. Thus, segmentation by texel similarity in the absence of curvature and perspective (requiring planar objects and orthographic projection), has been explored by, among others, [Tomita et. al., 1973]. Single simply-curved surfaces which fill an entire image obviate segmentation; if orthographic, they isolate the problem of determining a small number of global shapes from local clues, as in the analogous intensity work of [Woodham, 1977]. Single simply-textured planar surfaces (e.g. checkerboards) can effective isolate the last aspect, perspectivity. Perspectivity has received little attention. In fact, much research assumes orthography, and takes pains to compensate for, rather than utilize perspective effects.

Additionally, independent of these considerations are those shape and textural effects arising from the definition and arrangement of the texture components themselves. Clearly, structural textures are "easier" than statistical ones. Further, consider the already mentioned dichotomy of "painted" versus "pointed" textures: that is, the distinction between two- and three-dimensional texture components. Any analyses of the latter case is greatly complicated by the three-dimensional perspective transformations of the components themselves, and the associated effects of occlusion, mutual illumination, and shadowing. (Note, however, that a three-dimensional component with a known, definite normal extent can be useful in disambiguating local microplane orientations.) Given the uncertainty arising from the loss of information in the projection, from the complexity of the shape-from-texture phenomena, and from the infinite range of texel types, it appears that the coordination of surface integrity, orientation, and position is a heuristic task of artificial intelligence dimensions.

A New Aggregation Transform

Suppose then that the task of determining shape from texture is simplified to the following very simple subtask. Texture components are restricted to be two-dimensional; they are, in fact, forced to be line-like, organized into textures in a mesh-like fashion (a "structural" texture; somewhat like a piece of graph paper, except that line segments need not be contiguous, nor must they have a fixed spatial frequency). The shape phenomena are restricted to perspectively alone. Thus, the scene is limited to large, fairly regularly ruled planes set at various distances and orientations. (This abstraction is no accident; it is an idealization of Carnegie-Mellon's "downtown Pittsburgh" task.)

These restrictions suggest several exploitable properties. Planarity implies that local orientation is global as well; the determination of local vanishing lines can be done once, in the large, with consequent improvement in accuracy. Segmentation is eased by the uniformity of texture component direction. The texels themselves are easily identified by an edge detector; no local region growing, etc., is necessary to define them.

The major problem, then, is to aggregate the texels (in this case, edgels) into surfaces, mindful of the vanishing points. Note that most traditional textural transforms are of limited use here. Most were developed with the implicit assumption that the image was the orthographic projection of a frontal two-dimensional scene. Thus, any attempt to aggregate which is based solely on their usually scalar measures would find it difficult to distinguish intrinsic textural variations from perspective-induced ones.

A new aggregation transform is motivated, then, by the desire to group texels according to the two or more vanishing points they orient towards. (This is a very strong condition. The occurance of two or more vanishing points is a special case of the general problem of the vanishing line, which exists independently of any texel orientation, and which occurs even with statistical textures.) Conceptually, it ought to be sufficient to extend an infinite line through each line segment, followed by the detection of accumulation points. Texels can then be classified into implied object surfaces by their orientation with respect to their vanishing points. In effect, this implements the general image understanding heuristic that converging image lines arise from parallel lines defining a surface within the scene.

Practically, the problem is a bit more complex. Many times vanishing points are very distant, if not infinite. Further, a solution should be computationally efficient. Lastly, it would be beneficial if an aggregation operation grouped together like-oriented texels in an efficient and usable representation, so that their ensemble can be studied for, say, density, spatial extent, spatial frequency, etc.

The vector version of the rho-theta Hough transform is a likely starting point for such a transform. Recall that edge points are mapped under it into sine waves; edge vectors are mapped into points [Dudani et. al., 1977]. Lines are found by accumulation points in the Hough space. It is easy to see that parallel lines are indicated by accumulation points having the same theta value. Further, mutually converging lines lie on the sine curve which is the transform of their vanishing point. Unfortunately, the sines in the Hough space are difficult to detect. It is likely one would need a second application of the more general version of the Hough to do so: mapping each potential sine point into a curve in a second space, and detecting there accumulation points. The original aggregation, however, does have the advantage of implicitly representing and aggregating like-oriented edge segments by exactly one curve.

The following modification of the vectored rho-theta Hough presevers its local grouping property, but represents aggregates in a form more amenable to detection. In addition, it is also computationally cheaper, conceptually and visually more forthright, and can be used to replace the other Hough-like transforms used for vector grouping (for example, in the gradient intensity transform method ("GITM") of [Fennema et. al., 1978]).

The basic new idea is to plot the rho-theta transform space on polar coordinates. This has many desirable effects. Points now map into circles which pass through the origin. Edge vectors still map into points; but now the position of the transformed vector with respect to the transform origin is parallel to the direction of the edge vector itself. Its distance from the transform origin is such that if the two spaces were superimposed, the transformed vector is on the line determined by the edge itself. These two properties make the transform easier to view and imagine. As an elegant bonus, no trigonometry is required to calculate it. If the edge vector is the vector $E = (E_x, E_y)$, and if its position in the image is considered the vector $P = (x, y)$, then the transformed point, represented as the cartesian ordered pair $T = (i, j)$, is:

$$T = ((E \cdot P) / \|E\|^2) E$$

where "·" is the dot product and "|| ||" is the Euclidean norm.

Further, the transform maps each set of mutually converging lines into a circle passing through the origin; the vanishing point is represented by that point on the circle farthest from the origin. In the degenerate case of parallel lines (infinite vanishing point), the transform is a line through the origin perpendicular to the parallels. Local grouping is preserved; but now the aggregate representation, is easier to detect directly, as seen below. (Almost all of the above discussion, including the computational efficiency, has been shown to apply analogously to other uses of the vectored rho-theta Hough. As an example, in the GITM method, what were once secant curves become straight lines).

In this particular application, the detection of circular arcs (that is, of transformed line aggregates) can be done efficiently in the following manner. Consider a second transform that involutes the radii (rhos) of all the transformed points. That is, all transformed edge vectors are taken from (rho, theta) into (K/rho, theta), for some K. This transform also has desirable effects. Infinite vanishing points are mapped into the origin. Lines through the origin (that is, the transform of parallel lines) are unchanged in direction. Most importantly, all circles passing through the origin (that is, the aggregation of transformed converging lines) are mapped into straight lines. The distance of these lines from the newest origin is inversely proportional to their corresponding vanishing points' distances; their normals parallel the vanishing point direction. Another bonus: if this transform is composed with the first, the combined operation is even cheaper that the first alone:

$$T = (K / (E \cdot P)) E$$

Lines, of course, are easy to detect; one pass of a line detector with one more level of the new polar vector rho-theta Hough is sufficient. Note that because these line aggregates are so simple, detecting them also is, especially when compared to the original suggesting of searching for sines.

Examples

The complete process is summarized and illustrated by the following figures, using both a synthetic image (Fig. 1a) and a portion of a natural scene which includes a building face (Fig. 1b). Edge vectors (Figs. 2a and 2b) are mapped into points in a polar rho-theta Hough space. Mutually converging lines are thereby mapped into circular arcs which pass through the origin (Figs. 3a and 3b). Involuting the transform space maps the arcs into straight lines (Figs. 4a and 4b); this step can be derived directly from the edge image. The detected lines (Figs. 5a and 5b) are mapped by a second application of the (non-involuted) polar aggregation transform into points. These points correspond to the vanishing points in the image (Figs. 6a and 6b). No trigonometry is necessary, and it is never necessary to map a point into curve. Thus the computation is efficient, and given the intermediate representation, useful in analyzing and segmenting oriented mesh-like textures.

Conclusion

Determining shape from texture has many facets; the transform reported here is only one small one. As work continues, it is hoped that more insight into the various aspects of the phenomena can be made concrete in further observations, methods, and algorithms.

References

S. Ullman, "The Interpretation of Visual Motion," Ph.D. Thesis, Departments of Electrical Engineering and Computer Science, M.I.T., 1977.

D. B. Gennery, "A Stereo Vision System for an Autonomous Vehicle," Proceedings of the Fifth International Joint Conference on Artificial Intelligence, M.I.T., 1977.

B. K. P. Horn, "Understanding Image Intensities," Artificial Intelligence, Vol. 8, 1977.

A. K. Mackworth, "Interpreting Pictures of Polyhedral Scenes," Artificial Intelligence, Vol. 4, No. 2, 1973.

F. Tomita, M. Yachida, and S. Tsuji, "Detection of Homogeneous Regions by Structural Analysis," Proceedings of the Third International Joint Conference on Artificial Intelligence, Stanford, 1973.

R. J. Woodham, "A Cooperative Algorithm for Determining Surface Orientation from a Single View," Proceedings of the Fifth International Joint Conference on Artificial Intelligence, M.I.T., 1977.

S. A. Dudani, and A. L. Luk, "Locating Straight-Line Edge Segments on Outdoor Scenes," Proceedings of the IEEE Computer Society on Pattern Recognition and Image Processing, Rensselaer Polytechnic Institute, 1977.

C. L. Fennema, and W. B. Thompson, "Velocity Determination in Scenes Containing Several Moving Objects," Technical Report, Central Research Laboratory, Minnesota Mining and Manufacturing Company, St. Paul, 1977.

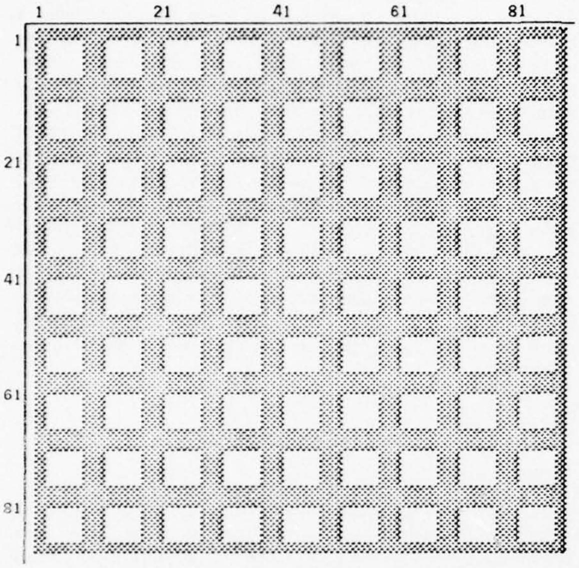


Figure 1a. Original (synthetic) image.

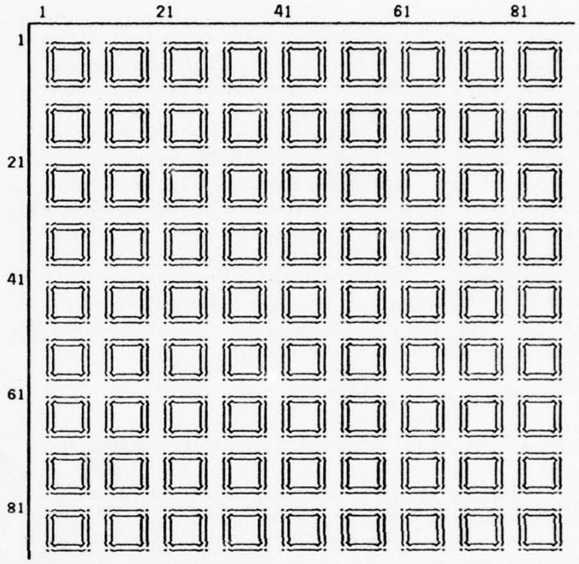


Figure 2a. Sobel edge operator applied to Fig. 1a.

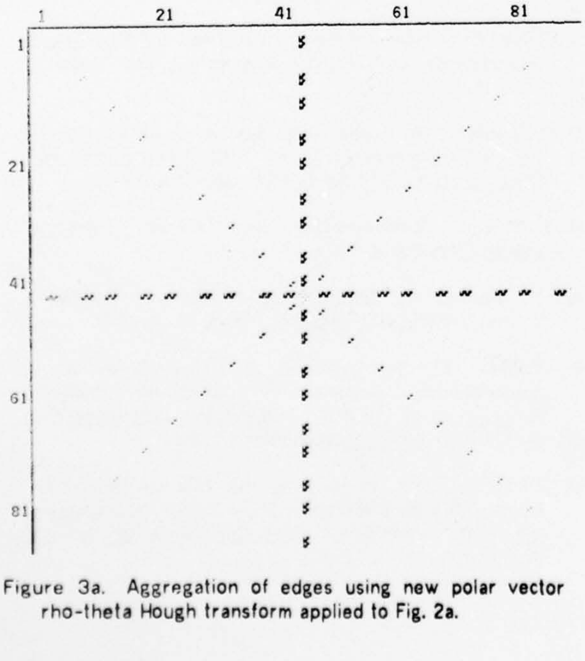


Figure 3a. Aggregation of edges using new polar vector rho-theta Hough transform applied to Fig. 2a.

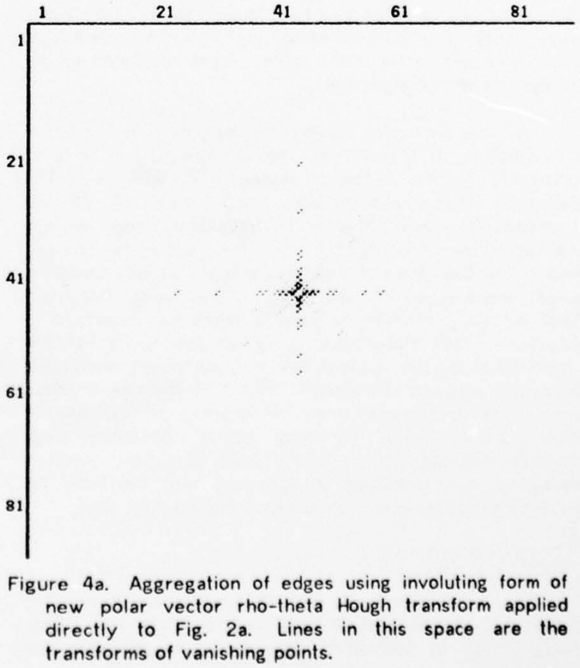


Figure 4a. Aggregation of edges using involuting form of new polar vector rho-theta Hough transform applied directly to Fig. 2a. Lines in this space are the transforms of vanishing points.

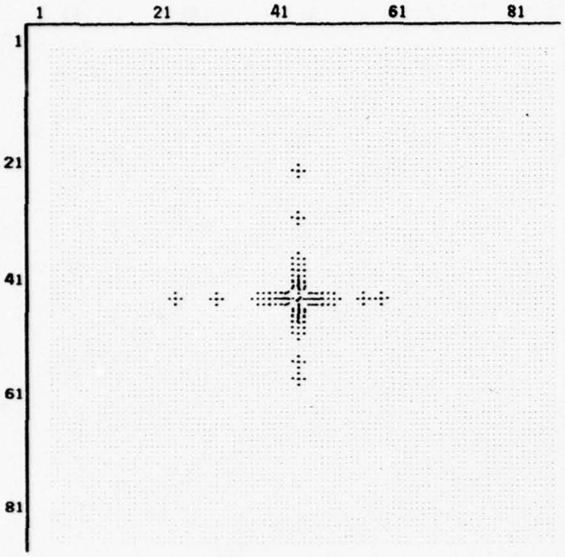


Figure 5a. Line operator applied to Fig. 4a.

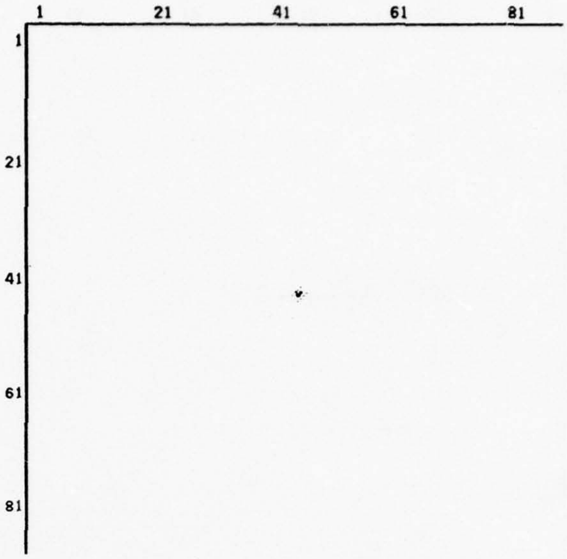


Figure 6a. Aggregation of lines using (non-involuting) polar transform applied to Fig. 5a.

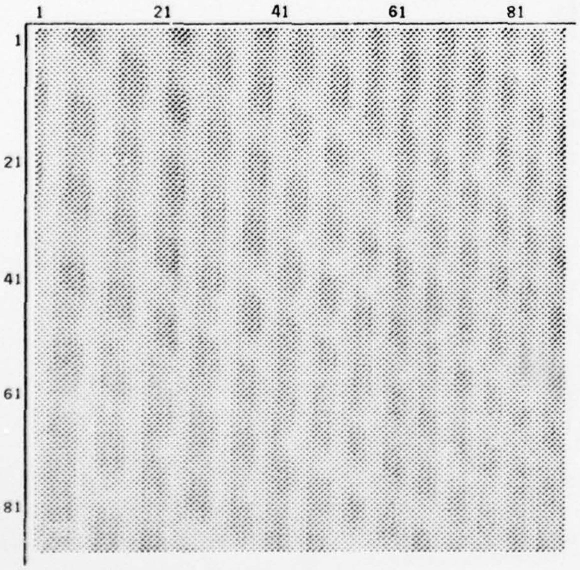


Figure 1b. Portion of natural scene (a building face).

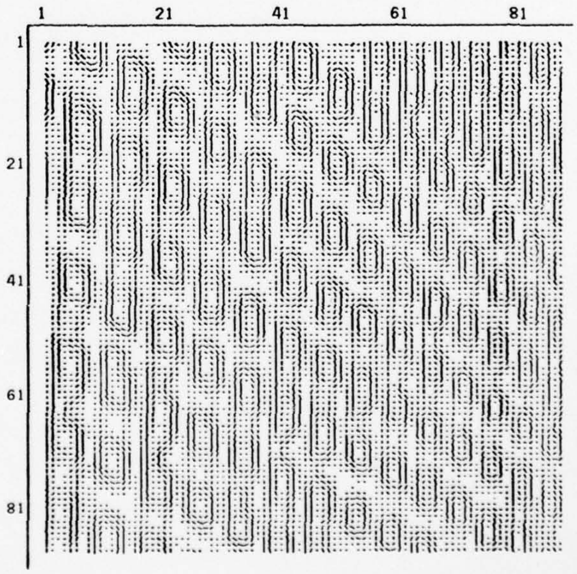


Figure 2b. Sobel edge operator applied to Fig. 1b.

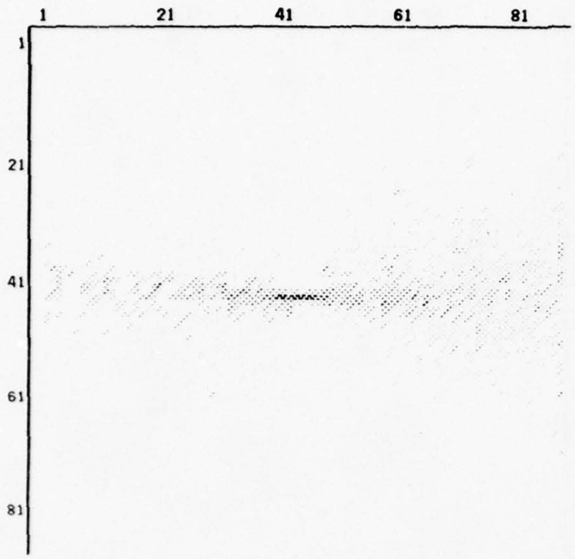


Figure 3b. Aggregation of edges using new polar vector rho-theta Hough transform applied to Fig. 2b.

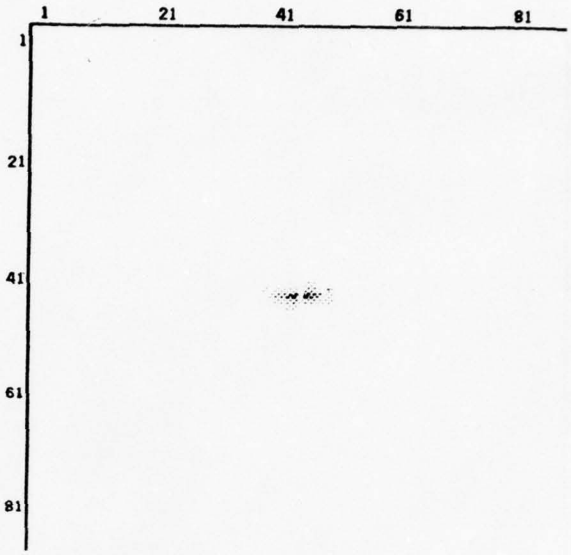


Figure 4b. Aggregation of edges using involuting form of new polar vector rho-theta Hough transform applied directly to Fig. 2b. Lines in this space are the transforms of vanishing points.

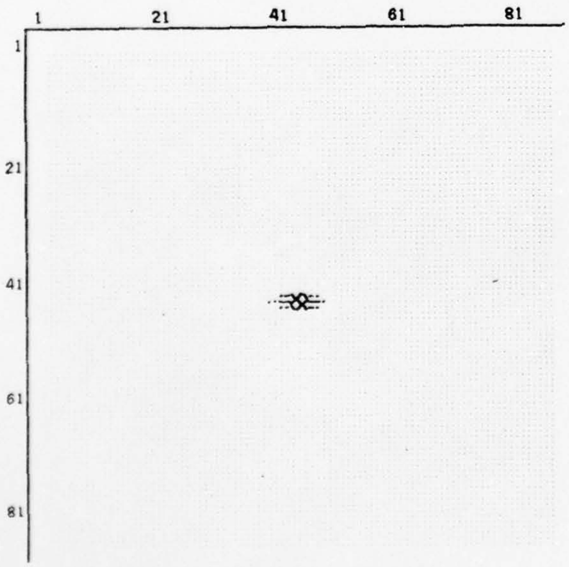


Figure 5b. Line operator applied to Fig. 4b.

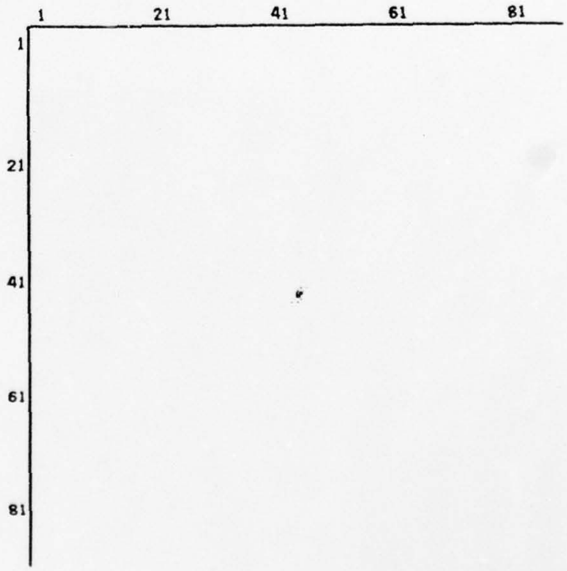


Figure 6b. Aggregation of lines using (non-involuting) polar transform applied to Fig. 5b.

EDGE POINT LINKING USING CONVERGENT EVIDENCE

David L. Milgram

Computer Science Center
University of Maryland, College Park, MD 20742

ABSTRACT

The thinned response of an edge detector constitutes a set of edge-points lying along edges in the original image. It is possible to link each edge-point to its appropriate neighbor on either side and thus delineate these edges in the image. This is accomplished by considering all contours produced by thresholding which pass through a given edge-point. For each such contour, the edge-point nearest the given edge-point along the contour in the clockwise direction is recorded. The edge-point appearing most often as clockwise associate to the given edge-point is then assigned as the clockwise neighbor. A figure of merit based on distance, straightness and contrast is used to break any ties. The counter-clockwise neighbor is computed similarly. The resulting weighted directed graph is available for segmentation into long chains, traversal, line-fitting or template matching. The use of contours to propose pairings of edge-points is an example of the power of convergent evidence.

INTRODUCTION

The importance of edge description is well documented by a rich literature. A large segment of the literature concerns the detection of points on region boundaries and the measurement of features, such as magnitude and direction, at those points. For a survey of edge detection techniques, see [1]. The edge points and feature measurements resulting from edge detection are put to many uses including threshold determination, segmentation, image matching, etc.

The grouping of edge points into higher order entities such as lines or curves has also received a good deal of attention. For an overview, see Section 8.4 of Rosenfeld [2]. Iannino and Shapiro [3] survey the Hough transform approach in which collinear points form detectable clusters and are thus associated into line segments. The sequential approach (tracking) attempts to extend the current line by affixing the best available edge point. Methods of this type are described by Montanari [4], Martelli [5] and Ashkar and Modestino [6]. A third class of methods is parallel in nature using, e.g., directed propagation to fill small gaps or relaxation to adjust incorrectly labelled edge points. See, for example, Zucker, et al. [7].

One may consider the problem of grouping edge points in a more general light. We might wish to group together those points which bound the same region in an image. However, the notion of "region" is imprecise due to conditions of poor lighting, shadows, non-planar surfaces, etc. If by a "region" we mean a "thresholdable region" then we may group together those edge points lying on the same contour after thresholding. This is the approach taken by Nakagawa and Rosenfeld [8]. However their pictorial results show that wrong associations are made when the assumption of region thresholdability is violated. The problem remains to associate edge points without requiring that the adjacent regions be thresholdable.

In [9], the author showed that the coincidence of edge points with region boundaries can serve as evidence for the presence of an object. In that method (called Superslice), an object might be evident over a range of thresholds and, for each threshold, might be represented by a different contour. Superslice selects the contour with the greatest percentage of coincident edge points. That approach relies on the convergence of evidence from two sources, thresholding and edge detection, to perform region extraction. The principle of convergent evidence is utilized in the current work to link each edge point to its best associate in the clockwise and counterclockwise direction. The algorithm which accomplishes this is called Superlink.

METHOD

To restate the problem: we are given a set of pixels (edge-points) corresponding to the locations and values of significant edge maxima in an image. Assuming that edge-points lie on edges extending some distance on either side, we wish to associate each edge-point to the appropriate edge-point on either side.

The solution is as follows: Let E be the set of edge-points and let $e \in E$. Suppose at some threshold τ, there is a connected component of above-threshold points whose boundary includes e. (We call all such boundaries "contours.") Let $e=e_1,e_2,\ldots,e_n,e_{n+1}=e$ be the succession of edge-points encountered in a clockwise traversal of the contour. We define $C(e,\tau)=e_2$ ($CC(e,\tau)=e_n$) as the clockwise (counterclockwise) neighbor of e. Each neighbor $C(e,\tau)$ delimits a path from e to $C(e,\tau)$ along a contour. At a different threshold τ', $C(e,\tau')$ might delimit a different path. We can compare paths preferring some to others based on various features and compute for each path a figure of merit. Thus, for example, short, straight paths are preferred as are those whose contrast does not vary much from one end to the other. The figure of merit used here is a weighted linear combination of length, straightness and contrast, although we recognize that many other possibilities and combinations exist. If no contour passes through e at a given threshold τ then $C(e,\tau)$, $CC(e,\tau)$ are undefined.

Consider the collection (including duplicates) of clockwise neighbors $C(e, \tau_1),\ldots,C(e,\tau_k)$ for some set of thresholds $T=\{\tau_1,\ldots,\tau_k\}$. Define C(e) to be the neighbor of e oocuring most often in the collection. Thus C(e) is chosen as the clockwise associate of e (the counterclockwise associate CC(e) is defined similarly). In the event that several edge-points occur equally often, choose as associate that edge-point contender with the highest figure of merit. It makes sense to delete any associate whose figure of merit is below some threshold, indicating the weakness of the evidence for linking the edge points.

When completed, the process has selected for each edge-point e (at most) one clockwise associate C(e) and (at most) one counterclockwise associate and has compared their figures of merit. Note however that the association is not necessarily mutual (symmetric), i.e. it is not true that $CC(C(e))=e$ or that $C(CC(e))=e$. This is reasonable since it is possible for an edge-point to be in the vicinity of a corner at which three or more surfaces meet. It may also result from breaking ties or from edge-point clustering. Nonetheless, the great majority of linkings do turn out to be mutual,providing additional evidence of their correctness.

IMPLEMENTATION

Superlink has been implemented on the Univac 1108 (Exec 8) and the PDP-11/45 (UNIX) as a sequence of modules described below (Figure 1). In the first step, edge-points are located in the input image by thinning the response of an edge detector. The detector we used computed the horizontal and vertical differences of 2x2 averages. The resulting difference images were separately thinned by local non-maximum suppression. The edge-point image results from taking the maximum of the thinned horizontal and vertical responses and deleting all insignificant edge responses (≤ 1).

Prior to contour extraction, it is necessary to determine the set of gray levels T at which to threshold the input image. Naturally, T can consist of the whole gray-level range in the input image. However, this can be expensive. Frequently, one has knowledge of the likely gray level range of the edge-points of interest. For example, in a two population image (object/background) the range between the modes would define T. Alternatively, one could sample the gray level range choosing every other gray level, etc. The danger in any scheme which skips gray levels is that all contours at the ignored gray level thresholds as well as the possible pairings of edge-points along these contours are lost. Thus less evidence is available when choosing associates, which makes the selections more dependent on figure of merit (a distinctly weaker criterion than "most often occurring edge-point"). Nonetheless, the degradation incurred by deleting gray levels is gradual as is discussed in the next section.

Once a set of gray level thresholds is chosen, the contours are extracted [10] and stored in Freeman chain code. This takes one pass over both images (gray level and edge-point) for each of the thresholds. The accumulated chain-encoded contours are stored on disk. Next, the disk file is read contour by contour. For each contour, the sequence of edge-points is noted and the figure of merit is computed for each adjacent pair in the sequence. The coordinates of each pair of edge-points and its figure of merit are then written to a file. The file of edge-point pairs is quite large (40,000 pairs for a 256^2 image using 12 thresholds). It is sorted (using a system sorting package) so that all pairs containing a given edge-point are in contiguous sequence.

The sorted file is a sequence of edge-point lists. Each edge-point list is the set of pairs for a given edge-point. Once it is read, it is straightforward to compute the most numerous associate or, in the event of a tie, the best figure of merit. The (edge-point, associate) pair is then written to a separate file. Finally, the associates file is converted to an image by taking each associated pair of coordinate pairs and drawing a straight line in the image to signify their linkage. This last step is convenient for display purposes; however, the use of a straight line to join edge-points only serves as an approximation to the contour segment which actually bridges the two points.

RESULTS

The algorithm as described in the previous section has been run using a variety of input images. Figure 2 shows several FLIR images of military vehicles and illustrates the extracted edge-points. Figure 3 displays the edge-points and their links. Links whose figures of merit were below a preset threshold are not shown. Our experience has been that selecting a threshold for the figure of merit is difficult unless a very generous one is used, as was done here. Normally, all but about 3% of the proposed links appear to be justifiable. Of course, some links are the result of more evidence than others and the figure of merit attempts to capture this. Thus the underlying data structure is a weighted directed graph, with the figure of merit corresponding to the weight.

A portion of an image of automotive parts from a GM data base [11] shows the effect of thresholding the figure of merit (Figure 4). As more links are deleted, some "obviously correct" linkages disappear while others which are somewhat more dubious remain. A blow-up of the upper-right portion (Figure 5) shows that where the edge-points form a staircase pattern, the linkages form small loops. Small loops may also result from the linkages of isolated points. This demonstrates as well that the process which creates edge-points must locate the points accurately, thin them sufficiently, and discard those deemed not to correspond to actual edges. Figure 6 shows the effect of thresholding the edge-point population on the linkages produced.

It was mentioned previously that the most effective linkages are produced when all gray level thresholds are employed but that degradation is graceful as gray-levels are omitted. Figure 7 illustrates the effect of retaining only every other gray level. Figure 8 shows another example of the GM data base along with the resulting linkages based on every other gray level.

CONCLUSIONS

The Superlink algorithm joins edge-points based on thresholding evidence. By and large, its proposed linkages are reasonable. Much work, however, remains. First, the current figure of merit, while well-founded, is heuristic and could benefit from further analysis. For example, no notice is taken currently of mutual linkages; yet, clearly, this is powerful evidence that the linkage is legitimate. Secondly, the choice of edge-points depends on the type of edge detector, the method of thinning and the elimination of noise points. Third, the steps making up Superlink can be consolidated and the processes made to run much more efficiently. Finally, new algorithms are needed to track the linkage data structure and to extract consistent boundaries.

REFERENCES

1. Davis, L.S. "A survey of edge detection techniques," Comp. Graphics and Image Proc., Vol. 4, No. 3, 1975, 248-270.

2. Rosenfeld, A. and A. Kak. Digital Picture Processing, New York, 1976.

3. Iannino, A. and S. Shapiro. "A survey of the Hough transform and its extensions for curve detection," Proc. of IEEE Conf. on Patt. Rec. and Image Proc., May 1978, Chicago, IL., 32-38.

4. Montanari, U. "On the optimum detection of curves in noisy pictures," Comm. of the ACM, Vol. 14, No. 5, 1971, 335-345.

5. Martelli, A. "An application of heuristic search methods to edge and contour detection," Comm. of the ACM, Vol. 19, No. 2, 1976, 73-83.

6. Ashkar, G. and J. Modestino. "The contour extraction problem with biomedical applications," Comp. Graphics and Image Proc., Vol. 7, No. 3, 1978, 331-355.

7. Zucker, S., Hummel, R., and A. Rosenfeld. "An application of relaxation labelling to line and curve enhancement," IEEE Trans. Computers, Vol. 26, 1977, 393-403; 922-929.

8. Nakagawa, Y. and A. Rosenfeld. "Edge/border coincidence as an aid in edge extraction," Univ. of Md. Comp. Sci. Ctr. TR-647, March 1978.

9. Milgram, D. "Region extraction using convergent evidence," Univ. of Md. Comp. Sci. Ctr. TR-674, June 1978.

10. Milgram, D. "Constructing trees for region description," Univ. of Md. Comp. Sci. Ctr. TR-541, June 1977.

11. Baird, M.L. "A computer vision data base for the 'industrial bin of parts' problem," Research Publication GMR-2502, Research Laboratories, General Motors Corp., Warren, MI, August 1977.

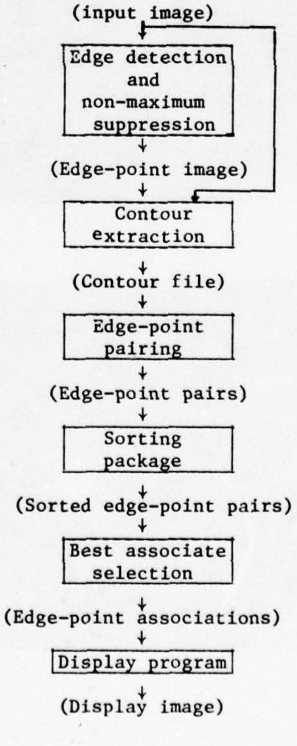


Figure 1. Superlink processing steps

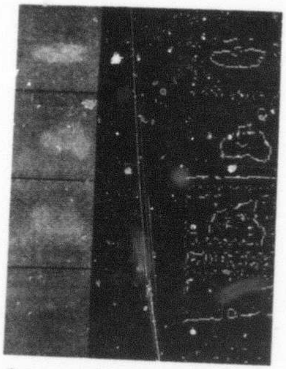
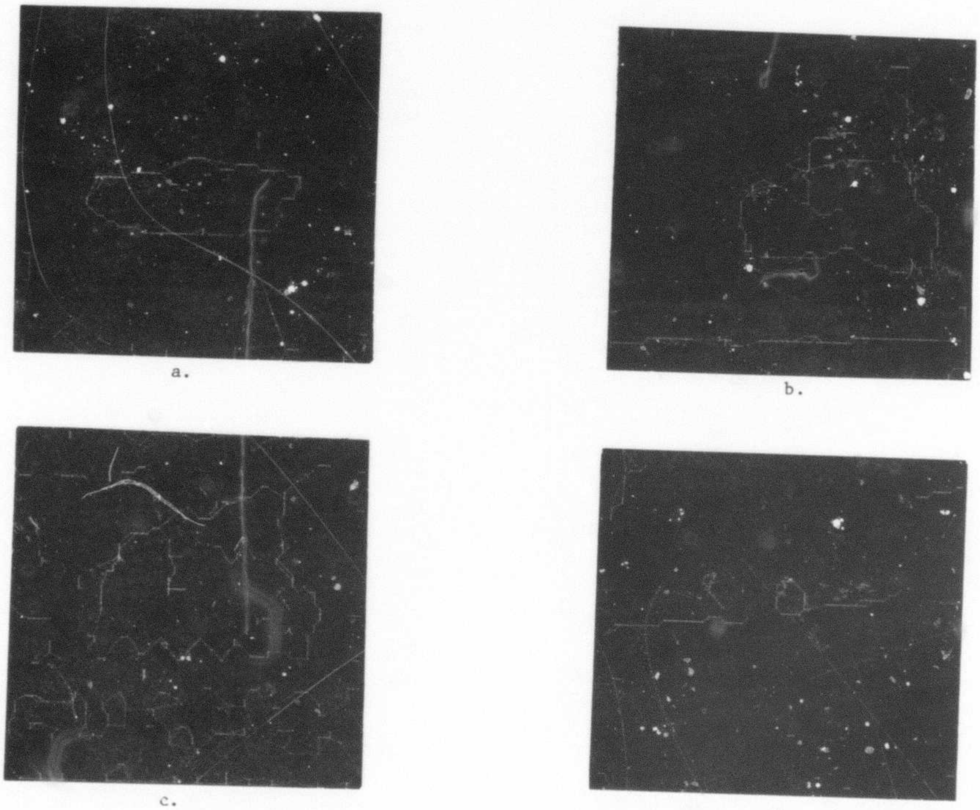


Figure 2. Gray scale windows and edge-point windows or four tanks from the NVL Data Base.



a.

b.

c.

d.

Figure 3. Edge-point associates of windows in Figure 2.

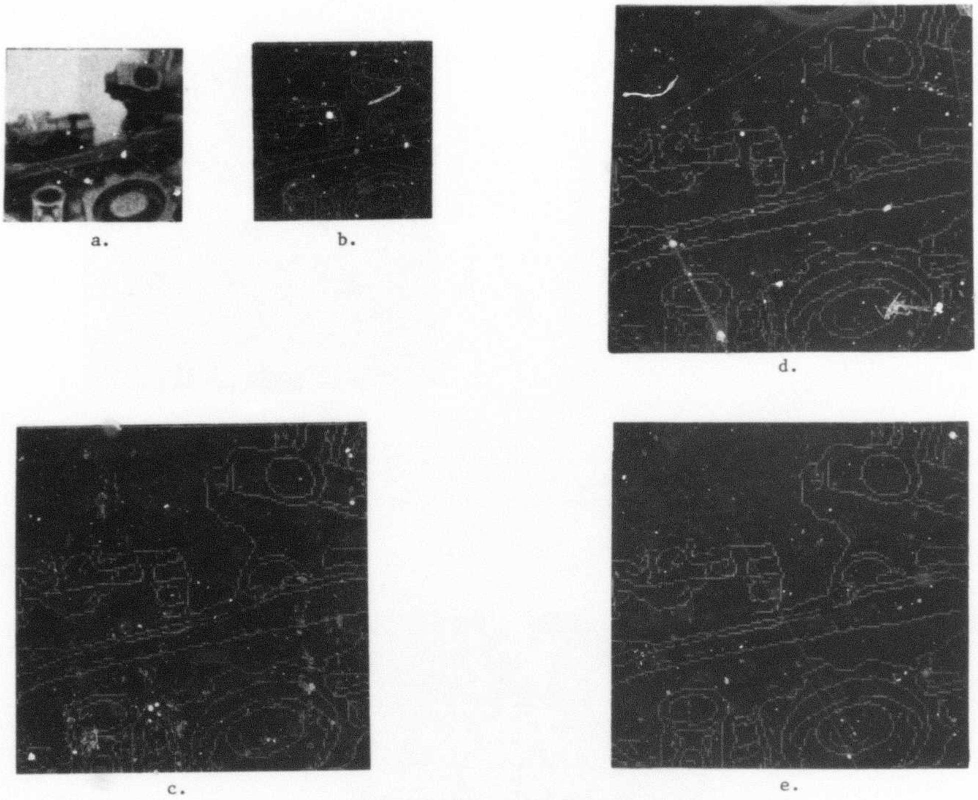


Figure 4. Effects of thresholding the figure of merit
a. Original gray level window
b. Edge-point image
c. Edge-point associates, threshold at .4
d. Same as c., threshold at .5
e. Same as c., threshold at .6

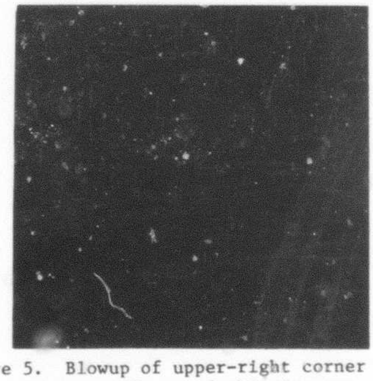
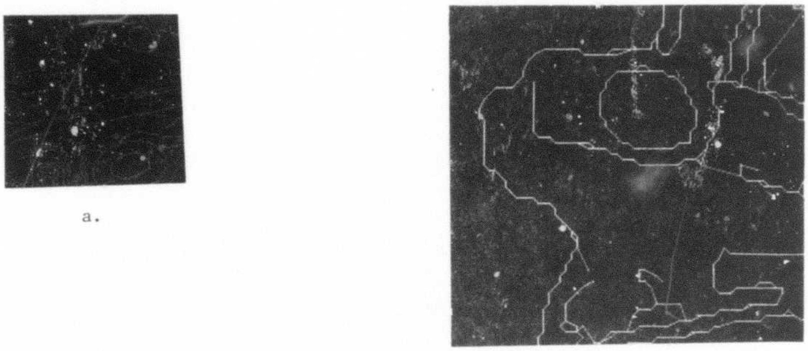


Figure 5. Blowup of upper-right corner of Figure 4c to display the effects of thinning errors



a.

b.

Figure 6. The effects of edge-point selection
a. Edge point subset of Figure 4b.
b. Edge point associates for upper-right quadrant

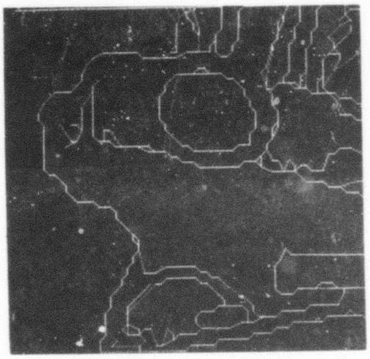
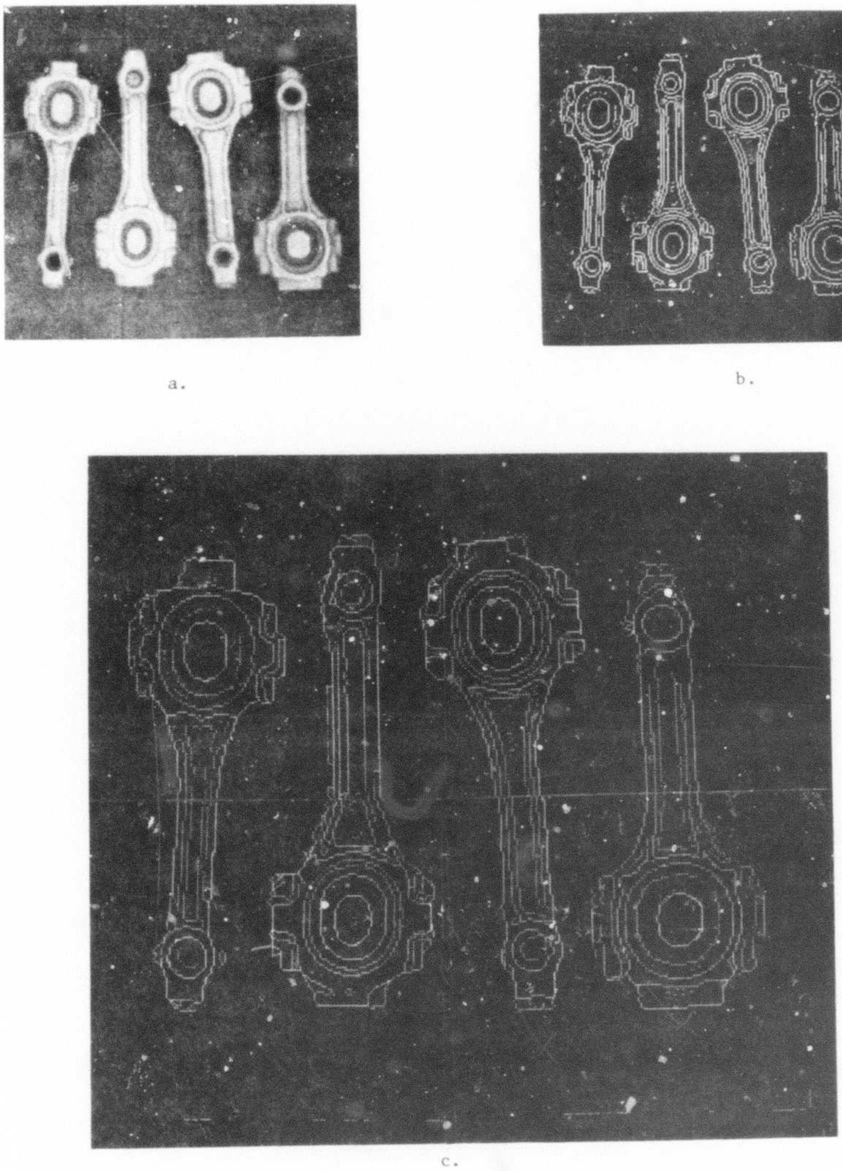


Figure 7. The effect of thresholding at even gray levels only. Compare with Figure 5.



a.

b.

c.

Figure 8. Another image from the GM Data Base
a. Original
b. Edge-point image
c. Edge-point associations

COMPUTATION OF LOCALLY PARALLEL STRUCTURE

Kent A. Stevens

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts U.S.A 02139

ABSTRACT

One computational problem in early visual processing is the description of local structure in the textured image. Of particular importance is parallelism, as arises in the images of fir, grass, wood, and so forth. Local parallelism in the three-dimensional world is generally preserved in the image, and therefore is a valuable structural property to describe. For instance, a portion of the image over which the texture is locally parallel would likely correspond to a single physical surface, regardless of mottled illumination variations (e.g., from partial shading) and variations in surface reflectance properties (e.g., camouflage) that would otherwise cause spurious region segmentations. Insight into a computational method for detecting this structure in an image was gained from a study of the human visual system's ability to detect local parallelism in dot patterns. A simple representation of locally parallel structure is proposed, and it is found to be computable by a non-iterative, parallel algorithm. An implementation of this algorithm is demonstrated; its performance parallels that observed experimentally (thus suggesting a potential explanation for human performance). The computation method generalizes to extracting parallelism in natural texture.

1. INTRODUCTION

A Moiré effect can be seen in patterns constructed by superimposing two copies of a random dot pattern where one copy had undergone some composition of expansion, translation, or rotation tranformations (figure 1a-1d) [Glass, 1969]. Our perception of structure in these "Glass patterns" has been taken as evidence that the visual system performs local autocorrelations [Glass, 1969; Glass & Switkes, 1976]. That is, the Moiré effect is due to the detection of pairs of correlated dots, each pair consisting of a dot in the initial pattern and the corresponding dot in the transformed copy.

Glass [1969] observed that the Moiré effect diminishes in the rotation-generated patterns as the amount of rotation increases. The periphery of the pattern (where the rotation causes the largest displacements) is the first to lose the circular organization. With sufficient rotation, one is left with an apparently random dot pattern. Furthermore, the Moiré effect will disappear if all but a small portion of the pattern is occluded [Glass & Perez, 1973]. Thus the effect is somehow dependent on the displacements between correlated dots, and the number of pairs of dots presented. The correlated dots need not be nearest neighbors for the effect to occur [Glass & Perez, 1973]. Recently it was shown that the pairs of correlated dots must correlate well in terms of orientation [Glass & Switkes, 1976]. In addition to detected parallelism, dots organized into chains and clusters contribute to the Moiré effect. Glass patterns can be constructed in which these latter contributions are insignificant, allowing a study of the detection of local parallelism.

This raises a number of interesting questions concerning (1) the representation of the parallelism, since there are no elements in the image with inherent orientation, (2) the means by which this representation is computed, and (3) why this structure is perceived. Prior to addressing these questions, the relationship between the perceived effect and the displacements between corresponding dots will be studied. Then, a method will be introduced for computing a representation of this structure. Finally, a use for this representation is suggested.

2. EXPERIMENT

The experiment studied the effect of increasing the displacement between corresponding dots on the detection of parallelism among dot pairs. The goal was to determine the maximum tolerable displacement as a function of the dot density.

2.1 Method

2.1.1 Glass Patterns

The patterns consist of two superimposed copies of an initial dot pattern. Glass and Perez [1973] used random dot patterns. However, the use of random dot patterns confounds the Moiré effect with clusters, sparse regions, and especially, chains of dots. When the transformed copy is superimposed, these inhomogeneities are selectively enhanced, and provide strong clues as to the transformation that was applied. Relative to the initial pattern, each dot in the transformed copy is displaced along a trajectory. If N dots in the initial pattern are aligned such that they would be displaced along a common trajectory, then there would be a chain of 2N dots after the second copy is transformed and superimposed. Thus even two adjacent dots, if they happen to be so aligned, will cause a conspicuous chain of four dots. For expansion or rotation transformations, the chains would then be radial or concentric, respectively. Those boundaries of clusters and sparse regions in the initial pattern that happen to align with transformation trajectories are similarly enhanced. Consequently, clusters and sparse regions that appear amorphous and randomly oriented in the basis

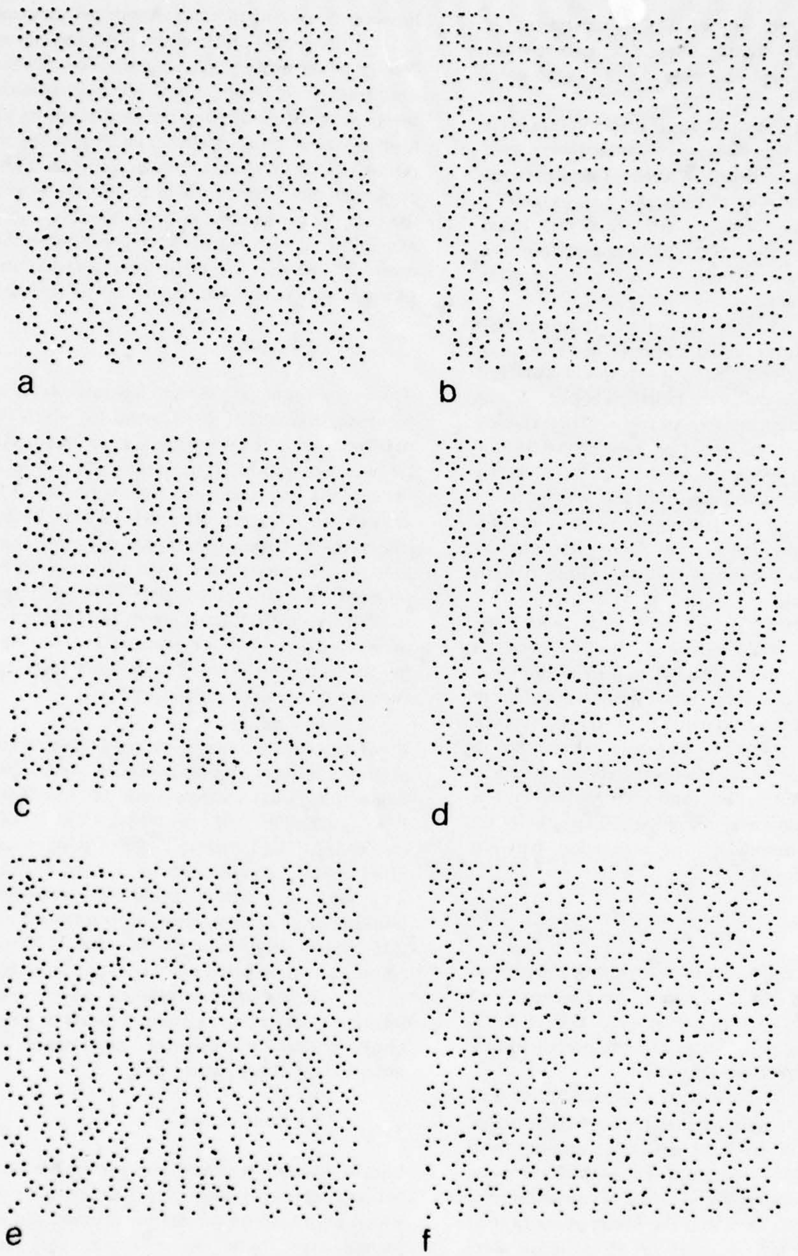


Figure 1. Glass patterns constructed from a pseudo-random dot pattern and a superimposed copy of that pattern which has undergone some homogeneous displacement transformation. The patterns contain approximately 800 dots (ρ= .0124). The translation, spiral, radial, and concentric patterns (figures 1a-1d) all have displacements between corresponding dots of 7.7 units (pattern dimensions are 256 by 256 units), N=1.95 (number of extraneous neighbors lying nearer to a given dot than its corresponding dot). Figure 1e is a composite pattern composed of portions of the patterns in figures 1a-1d. The local structure is seen to be independent of the global organization. In figure 1f the radial effect is due to chains, not pairings between corresponding dots (displacement=10.0 units, N=3.75)

pattern appear wedge shaped in radial Glass patterns, or crescent shaped in rotational patterns. These clues persist when the transformation is so extreme as to make the correlated pairs indiscernible (figure 1f).

To reduce the effects due to clusters and sparse regions in the basis patterns, pseudo-random patterns were used in which the dots were more evenly distributed. These patterns were constructed by randomly perturbing the positions of a regular grid of dots. Chains would still arise, however, unless care was taken to generate the initial pattern knowing the *transformation that would be applied, so that adjacent dots* would not lie along a common trajectory.

Radial patterns without subjective chains were constructed by computing a basis pattern of randomly positioned dots on virtual spokes. Each spoke would hold one dot, thus insuring that no two dots were radially aligned. It was also important to avoid chains between nearly radially aligned dots. Therefore, to determine the radial position of the dot for each spoke, random values were computed and compared to the radial positions of the previous few dots until one was found to be sufficiently separated from its neighbors. The minimum allowed separation and the number of prior dots to be examined were empirically chosen so that the Glass pattern presented no subjective chains.

The Glass patterns were constructed with the corresponding pairs of dots separated by a constant displacement ("homogeneous displacement"), instead of the more natural "differential displacements" that would arise from rotation or expansion of the whole pattern. In the latter case, the displacement would be a function of the radial distance to the center of rotation or expansion. Homogeneous displacement patterns produce strong Moiré effects, and offer the advantage that since the effect is uniform over the entire pattern, the effect also tends to vanish uniformly as the separation between corresponding dots is increased.

2.1.2 Presentation

Sequences of homogeneous displacement patterns were presented to six unpaid volunteer graduate students. All patterns were presented on a Digital Equipment Corporation GT-44 CRT display in a darkened room on a 23.5 by 23.5 cm. screen from a distance of 115 cms. (11.5 degree visual angle).

In the following, the dot density ρ = (number of dots in pattern) / 256^2. The first series of presentations consisted of chainless radial patterns of five dot densities ranging from ρ=.00298 (195 dots) to ρ=.00884 (580 dots). For each dot density, 8-10 patterns were constructed with a range of displacements (between corresponding dots) for which the Moiré effect ranged from obvious to inapparent. A total of 45 patterns were presented in randomized order, in three sequences of 15 patterns each. Each sequence was viewed three times by each S, with the S instructed to judge each pattern numerically: "0" if the pattern appeared unstructured, "1" if the dots appeared to be paired, "2" if the pairings were locally parallel (i.e., while fixating a pair of dots, the neighboring dots also appeared paired and aligned with the fixated pair), and "3" if the parallelism appeared particularly strong. They were encouraged to sample several places on each pattern (avoiding the center and extreme periphery) before making their judgement, and to interpolate between these values according to the appearance in those localities. The presentation time was open ended, however Ss usually took 3-5 seconds per judgement.

A second series of presentations consisted of very low density patterns (ρ=.00096, 65 dots). Four types of patterns were used (radial, concentric, spiral, and translation). For each type, seven patterns of differing dot displacements provided obvious to inapparent Moiré effects. The 28 patterns were presented in randomized order as a single sequence. The sequence was presented three times to each S, and the S was asked to judge the patterns in the same manner as before, and to name the type of pattern as well. A typical response would have been "1.6 R" meaning "the dots appear paired, moreover in most places the pairings appear aligned; the overall pattern is radial."

2.2 Results

The responses of each S were separately tabulated, and for each sequence, that *critical displacement* for which the locally parallel pairings were just perceptible (i.e., an interpolated judgement of 1.5) was determined. The mean critical displacement for each density was then computed (see figure 2a). The data in figure 2a can also be expressed as follows: Define D to be the displacement between corresponding dots (constant across the pattern). Then consider a circular neighborhood of radius D centered on any given dot. The corresponding dot lies somewhere on the circumference of that circle. The number of other dots that would be expected in that neighborhood (i.e., to lie closer to the given dot than its corresponding dot) is a function of the dot density, specifically

$$N = \rho\pi D^2.$$

Figure 2b shows a plot of N versus density computed from the averaged critical displacements of figure 2a. The mean N values for radial patterns were 2.31 (ρ=.00096), 1.91 (ρ=.00298), 2.33 (ρ=.00443), 2.37 (ρ=.00587), 2.36 (ρ=.00739), and 2.36 (ρ=.00884). The mean for ρ=.00298 is significantly less than the other means, as indicated by a t-test ($p<0.05$, $t=2.83$, d.f.=32). The very low density (ρ=.00096) translation and concentric patterns resulted in insignificantly different means (N=2.40 and 2.31, respectively), however the critical displacement for the *spiral* pattern occurred early, resulting in N=1.68.

Follow-up presentations using various densities of translation, spiral, and concentric Glass patterns have shown the same critical displacement dependency on dot density, independent of the pattern type.

2.3 Conclusions

Locally parallel structure was perceptible until the separation between corresponding dots reached a critical displacement, which depended on the dot density, and did not depend on the pattern type (with one exception: very low density spiral patterns). The results can be interpreted as follows: *if more than two or three dots lie closer to a given dot than its corresponding dot, then locally parallel structure among such dots cannot be perceived.*

This is a statement about the limiting geometry in the patterns. In arriving at this result, a neighborhood was defined, whose radius was equal to the critical displacement. This neighborhood is merely a means for describing the local geometry of the dot patterns, and is not to be construed as some neighborhood used by the visual system in perceiving these patterns. Later, a computational neighborhood will be introduced.

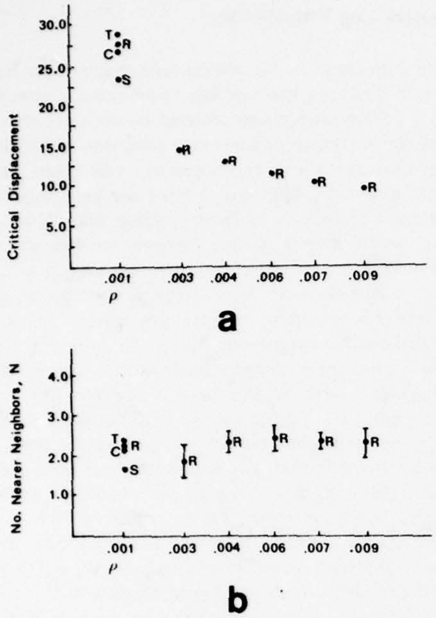


Figure 2. For a given dot density, there is a critical displacement (between corresponding dots) beyond which pairings between these dots cannot be perceived. This critical displacement (associated with an interpolated judgement of 1.5) was determined for each S, for each density. In figure 2a, the mean of these critical displacements is plotted as a function of dot density. Points in figure 2a replotted in terms of number of extraneous, nearer neighbors. Translational=T, radial=R, concentric=C, and spiral=S. Each vertical bar indicates two standard deviations.

For dot density ρ=.00298, the critical displacement occurred early This trend was recognized as the experiment was performed, and discussed with each S directly after the experiment. Their comments suggest the following interpretation. The initial presentations consisted of randomized sequences of patterns with five dot densities (ρ=.00298 through .00884). Relative to the higher dot densities, those of ρ=.00298 appeared less "locally parallel" for there were subjectively far fewer dots presented. There was apparently some coupling of the evaluation of locally parallel with the number of pairs that could be evaluated. However, in the second series of presentations, involving only patterns of ρ=.00096, the Ss appeared to be unaffected by the small number of pairs presented. The results with this dot density were in close agreement with the N=2.3 relation observed for the higher densities, with the following exception.

The critical displacement for spiral patterns of ρ=.00096 was relatively small, resulting in N=1.68. Comments from the Ss revealed that while the pairings could be held for relatively large displacements (i.e., sufficient to achieve N > 2.0), the pairs were not seen as locally parallel. However, since the spiral patterns were comprised of only thirty or so pairs of dots scattered over the display, one would not expect the widely separated pairs to appear locally parallel. At least with concentric and radial patterns some of the neighboring pairs relative to a given pair will be parallel. For example, with a concentric pattern, those pairs that lie on the same (or a nearby) radius will be approximately parallel. In fact, the results with very low density radial, translation and concentric patterns were similar, and in close agreement with the results from higher density patterns.

The critical displacement is sensitive to the *local* dot density, for if a Glass pattern is constructed with varying dot density but constant displacement between corresponding dots, the effect is apparent only in those neighborhoods where N would be less than two or three.

Whatever computation we perform on these patterns, it is relatively independent of the actual dot density. If a local computation is involved, then the angular extent of the neighborhood is determined by the measured dot density in that locality. Before arguing that the computation is local, one further result should be mentioned.

There had not been any investigation into the time required to perceive the structure in these dot patterns. In fact, it was not known whether eye movements are necessary for developing the impression of structure. To study this, a sequence of masking random dot patterns were presented before and after a single Glass pattern. The eight masking patterns had the same dot density as the Glass pattern, and the sequence was presented without pauses between frames. The frame rate was the experimental variable. It was assumed that in order to detect the Moiré effect, that the locally parallel structure would have to be determined within the time that the Glass pattern was presented. Thus the minimum presentation time would approximate the minimum computation time for determining the locally parallel structure. Note that once the local structure is determined, the global Moiré effect may continue to develop during the presentation of the subsequent masking patterns. It was found that at 80-90 msec/frame, one could reliably name the type of pattern. At 100-110 msec/frame one could name two different Glass patterns that were presented in succession while embedded in the masking sequence. Since the two patterns were presented in the same visual region, it is more likely that we perform two fast computations in sequence rather than two slower ones in parallel. Thus the computation of locally parallel structure is relatively fast, and does not require eye movements.

3. REPRESENTING AND COMPUTING LOCALLY PARALLEL STRUCTURE

Glass [1969] suggested that in our perception of these patterns, local correlations from different regions of the visual field are combined to form a simple global percept. That is, the processing is bottom-up, in contrast to the top-down alternative in which the overall structure is somehow determined, and that in turn influences the local percept. To support the bottom-up hypothesis, a composite Glass pattern (figure 1e) was created from portions of figures 1a through 1d. If the overall organization were to influence the perceived local structure, then one would expect that a neighborhood of dots taken from one pattern and embedded in another would appear differently in its new surroundings. However, the Moiré effect in any locality of figure 1e appears as it does in the original pattern (except along the boundaries where the neighborhoods have changed). The new global geometry does not influence the local structure.

It is easy to demonstrate that the Moiré effect requires a number of dots in order to be seen. If one masks out progressively greater portions of the pattern, the effect diminishes until so few dots are left that one becomes aware of coincidental arrangements among those dots [Glass & Perez, 1973]. If, however, the pattern is initially masked except for a few dots, and progressively larger neighborhoods centered on the initially visible dots are revealed, then the initial, coincidental groupings of dots are replaced by pairwise groupings. As the pattern becomes more fully exposed, those pairings remain, and are seen to be locally parallel. When our awareness is on the overall pattern, we see a Moiré effect, while under scrutiny, we see pairs of dots. Note that very close pairs of dots can also be seen that are oriented contrary to the Moiré structure in that vicinity.

Thus two subjective impressions can be studied: the Moiré effect, and the pairings of dots. It is hypothesized that the global structure (e.g., "spiral", "radial") is derived from the local pairings, and constitutes a later, distinct computational problem. This paper is directed towards the more fundamental problem, how the pairings are represented, and how that representation is computed.

3.1 Proposed Represention

A natural representation for a perceived local pairing would be a *virtual line*. Each virtual line would represent the position, separation, and orientation between a pair of dots. The proposed representation of the local structure is simple, being a discrete, spatial arrangement of virtual lines. The Moiré effect would then arise from this local structure. The strength of the effect would be dependent on the size of the population, the length of the virtual lines, and their collective geometry.

The orientation of the local structure is represented only at discrete points in the image. Would a continuous representation be necessary? Consider an analogy to the representation of depth from stereopsis. Discrete stereo disparity clues result in a perceived surface that is continuous (e.g., in random dot stereograms [Julesz, 1971]). The strong impression of depth that we assign to all points in the image suggests that underlying this percept is a continuous representation of depth. However, a continuous representation for locally parallel structure would not be appropriate, for there is no evidence that we attribute a sense of orientation to all points in the pattern.

3.2 Computing the Representation

The fundamental problem in computing the representation is to determine which groupings to construct, for in the vicinity of any dot there are many neighboring dots with which the given dot can be paired. We understand that the perceived pairings are between corresponding dots, and that these pairings are seen to be locally parallel. While the corresponding dots cannot be known *a priori*, the virtual lines that would connect them would be locally parallel. Therefore it is hypothesized that the following method underlies the computation:

(1) virtual lines are constructed from every dot to each of the neighboring dots, and
(2) those virtual lines that are locally parallel are selected.

3.2.1 Constructing Virtual Lines

The first step is to construct the virtual lines that radiate from each dot to each of its neighboring dots. This raises a question as to how large the neighborhood centered on each dot should be. Since the computational problem is to select one virtual line (that which extends to the corresponding dot) from each neighborhood, it would be optimal to have the neighborhood just large enough to include its corresponding dot. A larger neighborhood would merely include more extraneous dots, a smaller one would fail to take the corresponding dot into consideration. Since there is no *a priori* knowledge of the position of the corresponding dot for any given dot, that neighborhood should be roughly circular.

The demonstrated independence of the Moiré effect from the angular extent of the pattern suggests that the neighborhood radius is a function of the local dot density. For now, consider that a neighborhood is defined on the basis of the local dot density, and that it is large enough to hold a few nearby dots. Better insight into the size of the neighborhood will be provided by the performance of an implementation.

Representing a small number of virtual lines that radiate from the center of the neighborhood poses no significant computational problems. In the proposed algorithm, a virtual line is represented by two quantities, an orientation, and a weighting. The weighting is greater for shorter lines, resulting in an algorithm that favors nearer pairings. This will be discussed in more detail later.

3.2.2 Selecting the Locally Parallel Lines

Given the virtual lines, the problem is now to extract those that are locally parallel. This problem can be solved simultaneously for each dot: that virtual line (from the given dot to one of its neighbors) which is parallel to the Moiré structure in the vicinity of that dot would be selected. Thus the problem, relative to a given dot, is to determine the orientation of the structure in its vicinity, then to select that virtual line with similar orientation. Since these neighborhoods overlap, the solutions would be everywhere locally parallel.

Given that a virtual line is represented as a weighted orientation, then if each neighbor contributed its virtual lines toward a histogram, then the local orientation statistics could be gathered. Note that each neighbor will contribute one virtual line that is actually the solution for that neighbor, i.e., it connects that neighbor to *its* corresponding dot. Those particular contributions will be parallel, hence will produce a peak in the histogram, and indicate the orientation of the Moiré structure in that vicinity. Therefore the problem of selecting the solution virtual line for a given dot is solved by chosing that line with an orientation similar to that of the peak in the histogram.

There are two neighborhoods associated with this method: the neighborhood within which virtual lines are constructed, and the neighborhood over which virtual line orientations are histogrammed. In this study, the two neighborhood sizes were equated (i.e., in terms of the number of included dots). Some support for this restriction will be given in section 3.2.5.

The following algorithm is applied to each dot in order to select the locally parallel virtual line for that neighborhood (see figure 3).

(1) histogram the orientations of the virtual lines of its neighbors,
(2) determine the peak orientation from the histogram, and
(3) select that virtual line whose orientation is closest to the peak orientation.

The consequence is that parallelism, *if present* in the local arrangement of virtual lines, would be detected and represented by those virtual lines that are selected. While the algorithm is phrased in terms of histogramming and peak selection, a biological implementation would blur the distinction between (1) and (2).

3.2.3 Limitations Inherent in the Algorithm

There are two immediate limitations that should be mentioned. First, if the neighborhood radius is determined on the basis of the local dot density, then the algorithm will fail whenever the corresponding dot lies beyond the neighborhood radius. Could that immediately explain the critical displacement phenomenon that we exhibit? That is, does the neighborhood radius equal to the critical displacement, so that when the corresponding dot lies beyond the critical displacement, it also lies beyond the neighborhood radius, hence not considered by the algorithm? Probably not, for within the radius of the critical displacement there are only two or three neighbors, which would be an insufficient sampling from which to produce a histogram with a reliable peak.

The algorithm is also limited by the orientation resolution, both in the representation of the virtual lines, and in their summation into the histogram. To illustrate, suppose that each dot has N neighbors. Then the area under the peak in the histogram would be at most N, while the total histogram area would be N^2, distributed over M "buckets" (determined by the orientation resolution). For any given M, if N is sufficiently large, the peak will be submerged in the histogram.

The Gestaltists recognized that we tend to see rectangular grids as either columns or rows, depending whether the vertical or horizontal spacings are smaller, respectively. The algorithm shares this behavior when proximity weighting is introduced. Without this proximity metric, the interior dots would have four strong peaks, corresponding to pairings in the principal diagonal orientations, the vertical, and the horizontal. Since the nearer pairing orientations are emphasized in the histogram, then that peak contributed by the nearer pairings is emphasized, allowing that orientation to be selected. However, proximity weighting will also limit the algorithm. Suppose that the displacement between corresponding dots is such that there are several extraneous nearer neighbors. The virtual lines to these dots would be emphasized more than the virtual line to the corresponding dot. As this would occur to the virtual lines *in any vicinity*, the contributions from the locally parallel lines would be relatively less effective in producing a peak in the histogram. Therefore, as the number of nearer neighbors increases (i.e., the displacement increases for a given dot density), the peak will become less significant.

If the neighborhood radius is large relative to the curvature of the structure (e.g., near the center of a radial or concentric pattern, especially with low dot densities), then the notion of "locally parallel" breaks down. The peak in the histogram would broaden, and selection of the solution orientation would become less reliable. The experiment demonstrated that locally parallel structure is difficult to perceive in low density dot patterns were the curvature is considerable.

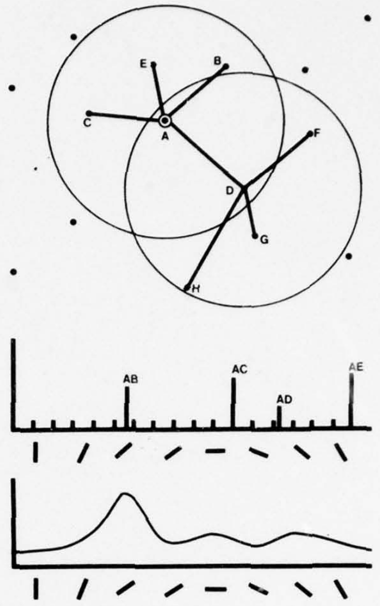


Figure 3. The algorithm has three fundamental steps: (1) construct virtual lines from each dot (e.g., dot A) to each neighboring dot (note emphasis of nearer neighbors); (2) histogram the virtual lines that were constructed for each of the neighbors; e.g., the neighbor D would contribute virtual lines DA, DF, DG, and DH to the histogram for dot A; (3) after smoothing the histogram, determine the orientation at which the histogram peaks and select that virtual line (AB) closest to that orientation as the solution.

In summary, the algorithm is fundamentally limited by three factors: the orientation resolution, the neighborhood size, and proximity weighting.

3.2.4 An Implementation of the Algorithm

An implementation in LISP has demonstrated that the algorithm is capable of computing the representation. The performance of the algorithm on various Glass patterns is demonstrated in figure 4, where the local orientation, as determined by the algorithm, is indicated by short line segments centered on the dots.

The virtual lines that radiate from a given dot to its neighbors were encoded by their orientations (the orientation resolution was 10 degrees) weighted in a simple manner by their length relative to the neighborhood radius, depending on whether the neighboring dot was nearer than a quarter, less than one half, or greater than half of the neighborhood radius. The weights were 1, 2/3, and 1/3, respectively.

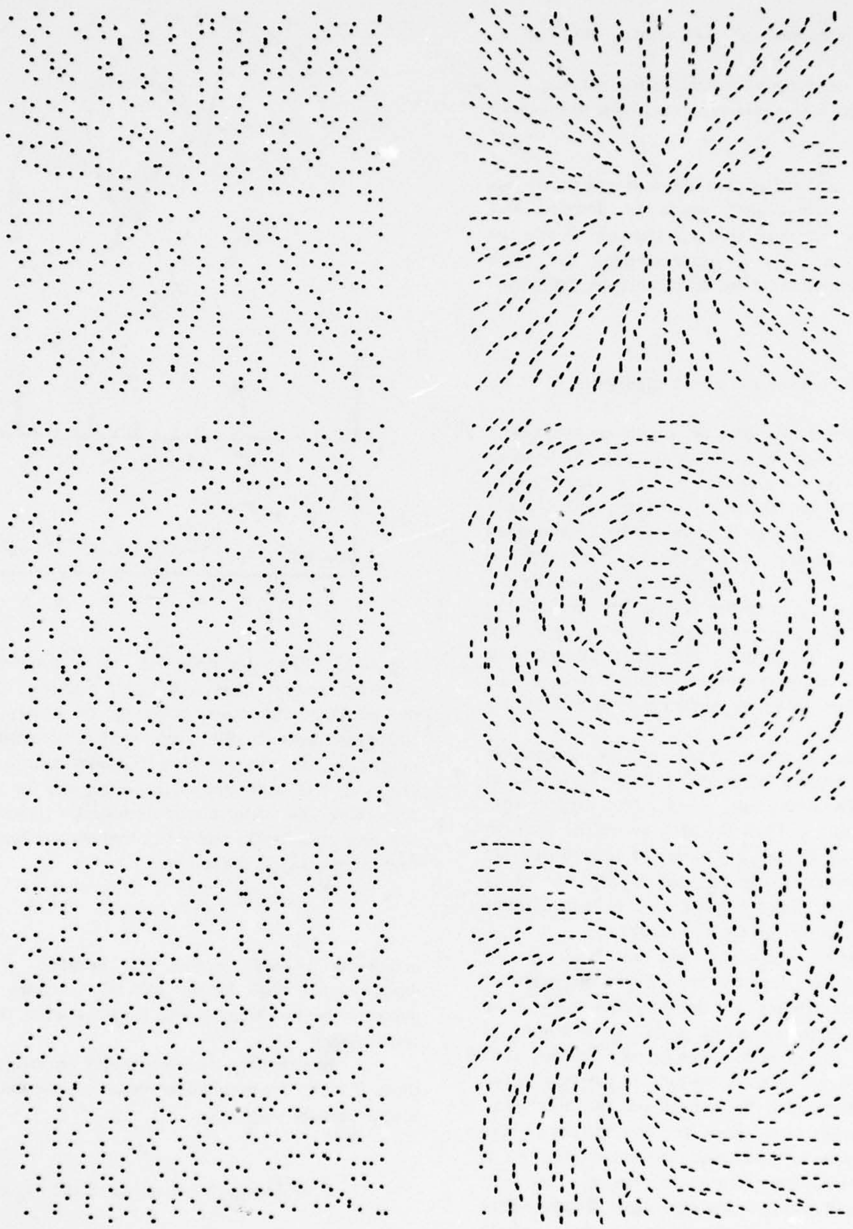


Figure 4. Demonstration of the algorithm on radial, concentric, and spiral Glass patterns (ρ= .0085; 556 dots; 7.7 unit dot displacement, therefore N=1.33). The algorithm used a neighborhood radius (20 units) such that roughly 8 neighbors were included. The solution at each dot is indicated by a short line segment.

The second step was to determine the solution orientation relative to each dot, computed by histogramming the weighted orientations associated with each of its neighboring dots and determining the peak orientation. Various criteria were studied for determining the peak of the histogram, with the conclusion that since the total area under the histogram curve is small, stringent criteria that require that the the peak be "significant" would often not be satisfied. With the exception of translation Glass patterns, the structure would not be strictly parallel in any neighborhood, causing the few contributions to the peak to be scattered over several adjacent histogram "buckets". Therefore a smoothing operator was applied to the curve to accentuate the peak, and that orientation with the maximum value was selected.

The final step was the selection of the solution virtual line from the set associated with each dot. That line whose orientation was nearest to the peak orientation was chosen and displayed graphically. If no virtual line was within 15 degrees of the peak orientation, then a dot was displayed, signifying that no solution was found.

3.2.5 Insight into our Critical Displacement Limitation?

If one were to accept the conjecture that we share the same algorithm for the perception of locally parallel structure, then could the LISP implementation provide us with insight into the cause for the observed limitations in our perception of the Moiré effect?

By varying the orientation resolution and neighborhood radius, the implementation of this algorithm can perform with either greater than or less than human ability (measured by the critical displacement between corresponding dots). An empirical study of this implementation was undertaken in order to determine if a particular choice of parameters would result in performance that closely matches ours. If that were found, then it would be interesting to reflect on the cause for the implementation's limitation given those parameters. Four orientation resolutions were used: 45, 33.3, 22.5, and 10 degrees (4, 6, 8, and 18 buckets). For each resolution, the algorithm was then run on translation and radial Glass patterns, while varying the neighborhood size. The first step was to increase the neighborhood size (measured by the number of included neighbors) until the performance was just breaking down at the critical displacement (N=2.36) while closely matching ours for lesser displacements. Then the algorithm was run (with the same neighborhood size) on radial patterns of various dot displacements, in order to verify that curvature does not effect the performance. It was found that reasonable performance could be achieved with as little as 33.3 degree orientation resolution when the neighborhood radius is such that only six or seven neighbors were included. This neighborhood radius is sufficiently small that curvature within that vicinity is insignificant, thus the performance is similar for radial and translation patterns.

If the histogramming neighborhood were significantly larger than the neighborhood for constructing virtual lines between neighbors, then one would expect human performance on translation patterns to be better than on spiral, radial, and other patterns with curvature. This follows from the peak contributions being precisely parallel in the case of translation patterns. However, the measured similarity in performance with translation patterns and those with curvature suggests that the histogramming neighborhood is not significantly larger than the virtual line neighborhood.

The conclusion drawn from this is that the parameter that governs the limiting performance is the neighborhood radius. Presumably, in choosing between (1) having a large sampling from which to make statistical decisions, and (2) restricting the area over which the samplings are taken, in order to avoid curvature, that the latter consideration is favored. The inevitable consequence then, is that the peak will often not be correctly distinguished from the noise. As discussed, proximity weighting helps when the corresponding dot is relatively nearby within the neighborhood, and hurts when it is near the perimeter of the neighborhood. When the corresponding dot is displaced by approximately 60 percent of the neighborhood radius (ratio of critical displacement to neighborhood radius) then the performance becomes significantly deteriorated.

While the performance is satisfactory with low orientation resolution, the performance with 10 degree resolution most closely parallels human performance. That is, if the solution line segments computed by the implementation do not correspond to the ideal solution in some small locality, it is often the case that we also perceive some anomolous groupings in that locality that are contrary to the overall Moiré structure. In summary, the algorithm exhibits human performance when the neighborhood is determined to be large enough to hold 6 or 7 neighbors, and the orientation resolution is 10 degrees.

4. HOW ABSTRACT ARE THE VIRTUAL LINES?

The proposed algorithm is based on virtual lines constructed between neighboring dots. The virtual line is an abstract construct that expresses a grouping between two elements in the image. Can a simpler explanation be found that would account for the Moiré effect, without having to construct some representation of groupings?

Glass [1969] suggested that the effect is evidence for local autocorrelation of the excitation of orientation-sensitive cortical units (presumably "simple cells" [Hubel & Wiesel, 1962]). According to this hypothesis, pairs of dots would tend to trigger these units when they happen to be aligned in their receptive fields. While the various coincidental pairings would result in the excitation of a large number of units, if their outputs were correlated over some neighborhood, the prominent orientation would correspond to the subjective flow orientation in that vicinity. Evidence that supports this hypothesis has been reported [Glass & Switkes, 1976].

However, there is some evidence to suggest that more is involved in our perception of parallelism in these patterns than simply the correlation of simple cell activity. Rival patterns will be described for which we prefer pairings between dots of similar intensity. Two consequences of this will be discussed: (1) that the Glass proposal does not correctly predict this preference, and that (2) we should consider the pairings as groupings between abstract places in an image.

Consider a Glass pattern constructed from the superposition of three patterns: an initial pattern, and two differently transformed copies. The resulting pattern is potentially rivalrous, for there are two locally parallel structures (figure 5a). First consider the case when the dots are of equal intensity and the displacements undertaken by both

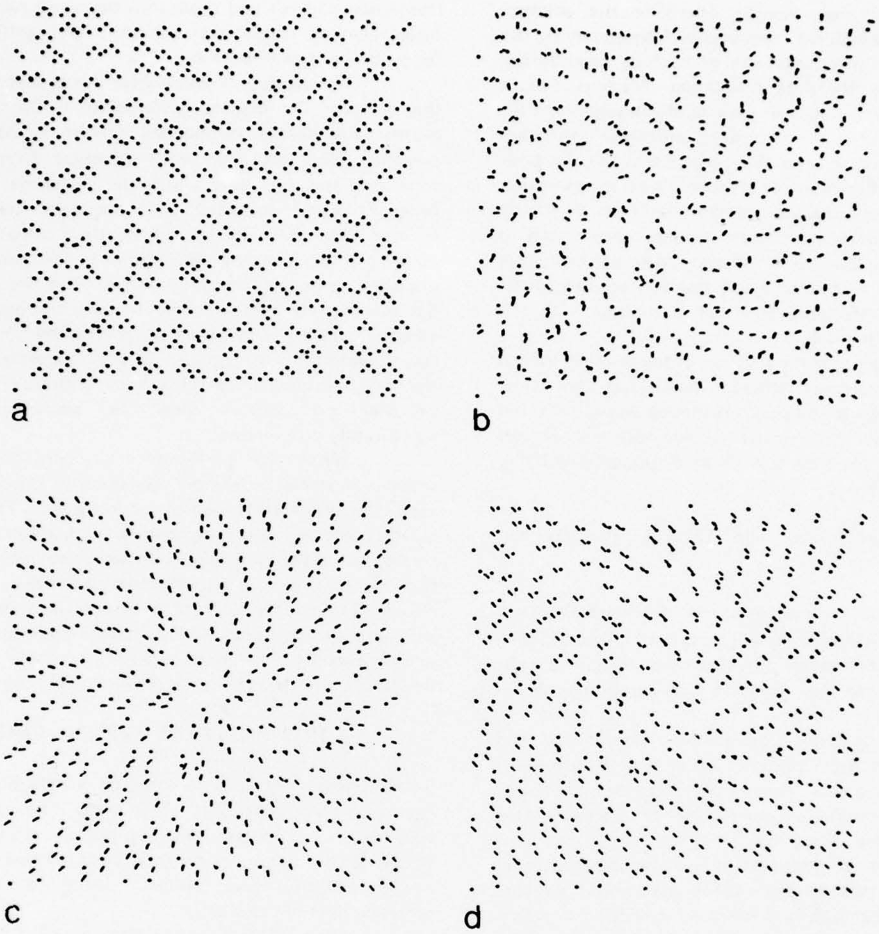


Figure 5. Rivalrous pattern (figure 5a), created by superimposing two differently transformed copies. In figures 5b-5d, a spiral Moiré effect is evident although derived from pairings between dots and short line segments. The lines are randomly oriented in figure 5b, while in figures 5c and 5d, the lines have global radial and translation organization, respectively.

transformations are equal. Locally parallel organization is difficult to perceive. However, with some effort we can extract either of the organizations, wherein the other (unpaired) dots are see as background.

Now, if the dots of the initial pattern and those of one of the transformed copies are displayed with low intensity, while the dots of the other transformed copy are of higher intensity, then we favor the organization consisting of pairings between low intensity dots. The subjective impression is one of parallel structure among the faint dots and a superimposed random pattern of bright dots. It is difficult if not impossible to perceive pairings between faint and bright dots as being locally parallel. If one fixates on such a pair, then the vicinity appears heterogeneous (i.e., to consist of *pairs* of faint dots mixed with individual bright dots).

The display apparatus gives us the facility to continuously vary the relative intensities of these two populations of dots. The display instructions specify two intensity levels, however, a potentiometer that governs the overall brightness can, in one extreme, make both intensity levels appear equally bright, while towards the other extreme make the lower intensity level effectively invisible while the higher level is still faintly visible. Thus all intensity ratios from 0:1 to 1:1 can be achieved. The rivalrous patterns appear ambiguous in the equal-intensity extreme (as in figure 5a). If one reduces the overall brightness, the lower-intensity dots become distinguishable from the higher-intensity dots, and pairings between the former are favored. In the extreme, these dots are so faint as to be insignificant, the brighter dots dominate, and the pattern appears random. At no point is there a preference for pairings between dots of differing intensity over those of like intensity.

It is difficult to account for this behavior with the mechanism based on correlated simple cell excitation. On the contrary, that proposal would predict the correlation to be stronger between faint-bright pairings, for units aligned with those pairings would be more excited that those oriented with the faint-faint pairings. What of the possibility that the faint-bright pairings do not enter into the correlation? One has merely to remove the competing faint dots in order to perceive a strong Moiré effect between the faint dots of the initial pattern and the bright dots of the remaining transformed copy.

It appears that some notion of similarity must be introduced into both proposed mechanisms. With the Glass proposal, the correlation must be on brightness as well as orientation and displacement (this may be difficult to provide with simple cells). Similarly, the histogram-based computation must introduce some notion of similarity. Clearly, one could introduce it in the same manner as proximity weighting (i.e., just as proximate dots are favored, so are dots of similar intensity). Then the virtual lines would express three quantities: the orientation, separation, and similarity between a pair of dots. This implies that dots should be considered as having at least one attribute other than position. Marr [1976] has introduced the notion of *place-token* as being a fundamental computational construct in early visual processing. It is essentially a means for attaching significance to a point in the visual field (such as the endpoint of some line or edge, or a dot [Marr, 1976; figure 12a]). These place-tokens are then the input to various processes that notice various relations in the local geometry of an image, which are then expressed as various *groupings* and *aggregations* [Marr, 1976]. The notion of place-token is supported here, for the locally parallel relation appears to arise from some computation that involves, not merely the local geometry, but other attributes of the image. These attributes would be associated with place-tokens. Marr suggests that place-tokens can be defined for midpoints of short line segments. It is interesting to note that we can derive a strong Moiré effect from patterns where, instead of dots, one is presented with dot-line segment pairs (figures 5b-5d).

5. DISCUSSION

A representation of locally parallel structure has been shown to be amenable to a particularly simple computation. The following issues have been illustrated:

(1) The computation is performed on place-tokens -- distinguished points that have been abstracted from an image.
(2) Virtual lines are constructed between pairs of neighboring place-tokens. The orientation and length of each virtual line is accessible to the computation.
(3) The orientation of the locally parallel virtual lines in any vicinity is determined by collecting local orientation statistics.

Why do we see the structure in these patterns? Two conjectures can be made, one with respect to motion, the other, about the general problem of seeing parallel structure in an image.

Glass and Perez [1969] found that if the relative intensities of the basis pattern and the superimposed patterns are dynamically varied, then apparent motion is perceived tangential to the Moiré, in the direction from lesser to greater intensity. They noted that the apparent motion differed from "phi" motion in two respects: (1) it requires a number of correlated dots in order to be seen (as does the Moiré effect), and (2) the corresponding pairs of dots must be simultaneously (rather than alternately) presented. If the Moiré representation were involved with motion, it would be useful for expressing correspondence relations between successive images. For example, if the initial pattern and the normally superimposed pattern are shown in succession, apparent motion can be seen. For this to occur, we must be establishing a 1-1 correspondence between dots seen in the first and second images. The proposed virtual line representation would then express this correspondence. The correspondence would be computed wholly on detected locally parallel trajectories.

This hypothesis that the algorithm computes locally parallel structure that expresses motion correspondence is weakened by the observation that the algorithm, while sufficient for the Glass patterns, is insufficient for pairing corresponding dots between frames of dot patterns, when the displacements undertaken by the individual dots between frames is considerable. As discussed, the algorithm tends to fail if more than roughly three extraneous neighboring dots lie closer to a given dot than its corresponding dot. However, if the two patterns that comprise a Glass pattern are presented in succession, then we can perceive rigid motion when an order of magnitude more extraneous dots (greater than 40) lie closer than the corresponding dot. To account for this ability, an algorithm based on histogramming would require very fine orientation resolution in order to detect the peak. It is probably unreasonable to expect that fine of orientation resolution in early vision. Furthermore, the emphasis placed on proximate neighbors, which is evident in the Moiré effect, is not apparent in the apparent motion effect (the dots appear to move as if attached to a rigid invisible surface, in spite of very near neighbors). A computation based wholly on the local geometry, as is this algorithm, would probably not be sufficiently constrained to solve this motion correspondence problem. Temporal and other constraints must be incorporated as well [Ullman, 1978].

The second conjecture concerns the perception of locally parallel structure in an image. According to this hypothesis, Glass patterns present stimuli to processes that (1) define place-tokens in the image, (2) construct virtual lines between neighboring tokens, and (3) extract those that are locally parallel. The algorithm by which (3) is accomplished is presumably applicable to "actual" lines and edges as well. Natural images often contain locally parallel textures (e.g., fur, grass, wood grain), which would result in large numbers of parallel line and edge elements in a description of that image. This structure could be extracted by a method based on

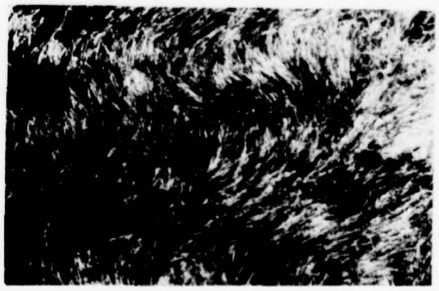


Figure 6. Photograph of human hair, example of local parallelism.

computing local orientation statistics, and selecting those lines and edges that are parallel to the prominent orientation in the vicinity. In figure 6 we perceive a certain homogeneity -- not of brightness, orientation, or line length -- but rather, of structure. That structure is locally parallel.

REFERENCES

Glass, L. 1969 Moiré effect from random dots. Nature 243, 578-580.

Glass, L. and Perez, R. 1973 Perception of random dot interference patterns. Nature 246, 360-362.

Glass, L. and Switkes, E. 1976 Pattern recognition in humans: correlations which cannot be perceived. Perception 5, 67-72.

Hubel, D.H. and Wiesel, T.N. 1962 Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J. Physiol. (Lond.) 160, 106-154.

Marr, D. 1976 Early processing of visual informantion. Phil. Trans. Roy. Soc. Lond. B. 275, 483-519.

Ullman, Shimon 1978 *The interpretation of visual motion.* Cambridge: M.I.T. Press.

QUANTITATIVE DESIGN AND EVALUATION METHODS FOR EDGE AND TEXTURE FEATURE EXTRACTION

by

William K. Pratt

Image Processing Institute
University of Southern California, Los Angeles, California 90007

INTRODUCTION

Quantitative design and performance evaluation techniques have been developed for image edge detectors and for texture feature extractors. The edge detector design techniques are based on statistical detection theory and deterministic pattern recognition classification procedures. The performance evaluation methods developed include: (a) deterministic measurement of the edge gradient amplitude; (b) comparison of the probabilities of correct and false edge detection; and (c) figure of merit computation. The design of texture feature extractors is based on stochastic field estimation. Evaluation is performed using a Bhattacharyya distance figure of merit.

LUMINANCE EDGE DETECTION

There are two basic methods of luminance edge detection: edge enhancement/ thresholding and edge fitting. With the former method, an image $F(j,k)$ is convolved with a set of N directional linear operators or masks $H_i(j,k)$ to produce a set of gradient functions

$$G_i(j,k) = F(j,k) \circledast H_i(j,k) \qquad (1)$$

where $\circledast$ denotes two-dimensional spatial convolution. Next, at each pixel, the gradient functions are combined by a linear or nonlinear point operator $\mathcal{O}\{\cdot\}$ to create an edge enhanced array

$$A(j,k) = \{G_i(j,k)\} \qquad (2)$$

Typical forms of the point operator include the root mean square, magnitude, and maximum. The enhanced array $A(j,k)$ provides a measure of the edge discontinuity at the center of the gradient mask. An edge decision is formed on the basis of the amplitude of $A(j,k)$ with respect to a threshold (t). If $A(j,k) \geq t$, an edge is assumed present, and if $A(j,k) < \bar{t}$, no edge is indicated. The edge decision is usually recorded as a binary edge map $E(j,k)$ where a one value indicates an edge and a zero value, no edge. Edge orientation can be determined from the compass direction of the maximum gradient function of eq. (1) or by the relation

$$\theta(j,k) = \tan^{-1} \frac{G_2(j,k)}{G_1(j,k)} \qquad (3)$$

where $G_1(j,k)$ and $G_2(j,k)$ denote the horizontal and vertical gradients, respectively. The most common edge detectors are listed below. Greater detail on their structure is found in Reference [1].

Differential edge detectors (N=2)

Roberts	(2x2 pixel)
Prewitt	(3x3 pixel)
Sobel	(3x3 pixel)

Template matching edge detectors (N=8)

Compass gradient	(3x3 pixel)
Kirsch	(3x3 pixel)
3-level mask	(3x3 pixel)
5-level mask	(3x3 pixel)

With the edge fitting class of edge detectors, image pixels within some region, typically 5x5 to 9x9 pixels, are fit to a two-dimensional step or ramp model of an edge. If the fit is close, an edge is deemed present and its parameters, bias, contrast, location, and orientation, are taken from the model. The most widely known edge fitting edge detector is the Hueckel operator [2]. Abdou [3] has recently developed another edge fitting operator with excellent performance.

EDGE DETECTOR SENSITIVITY ANALYSIS

Simple geometric calculations can be performed for the edge enhancement/ thresholding operators to determine the edge gradient response as a function of actual edge orientation. Results of these calculations are presented in Figure 1. The curves indicate that the Prewitt and Sobel square root differential operators and the template matching operators all possess an amplitude response relatively invariant to actual edge orientation. The Sobel operator provides the most linear response between actual and detected edge orientation.

STATISTICAL ANALYSIS

Edge detection can be regarded as a hypothesis testing problem to determine if an image region contains an edge or contains no edge. Let P(edge) and P(no-edge) denote the a priori probabilities of these events. Then, the edge detection process can be characterized by the probability of correct edge detection

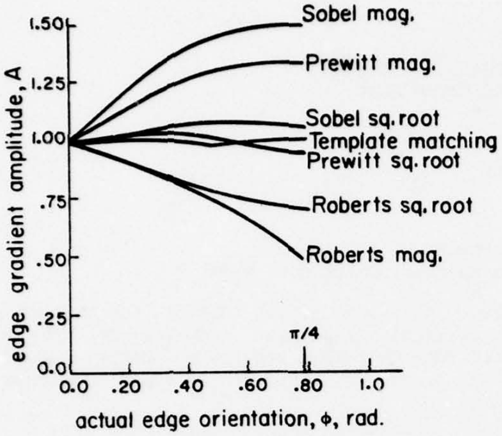


Fig. 1. Edge gradient amplitude response as a function of actual edge orientation for edge enhancement/ thresholding operators.

$$P_D = P(A \geq t|\text{edge}) = \int_t^\infty p(A|\text{edge})\, dA \quad (4)$$

and the probability of false edge detection

$$P_F = P(A \geq t|\text{no-edge}) = \int_t^\infty p(A|\text{no-edge})\, dA \quad (5)$$

where (t) is the edge decision threshold and p(A| edge) and p(A|no-edge) are the conditional probability densities of the edge enhanced field A(j,k).

The detection performance of edge detectors can be readily compared by a parametric plot of the correct detection probability P_D versus false detection probability P_F in terms of the detection threshold (t). Figure 2 presents such plots for square root differential operators and template matching operators for vertical and diagonal edges and a signal-to-noise ratio (SNR) of 10.0. From these curves, it is apparent that the Sobel and Prewitt 3x3 operators are superior to the Roberts 2x2 operators. The Prewitt operator is better than the Sobel operator for a vertical edge. But, for a diagonal edge, the Sobel operator is superior. In the case of template matching opertors, the 3-level and 5-level operators exhibit almost identical performance that is superior to the Kirsch and compass gradient operators. Finally, the Sobel and Prewitt differential operators perform slightly better than the 3-level and 5-level template matching operators.

FIGURE OF MERIT COMPARISON

The probabilities of correct detection and false detection, obtained analytically or

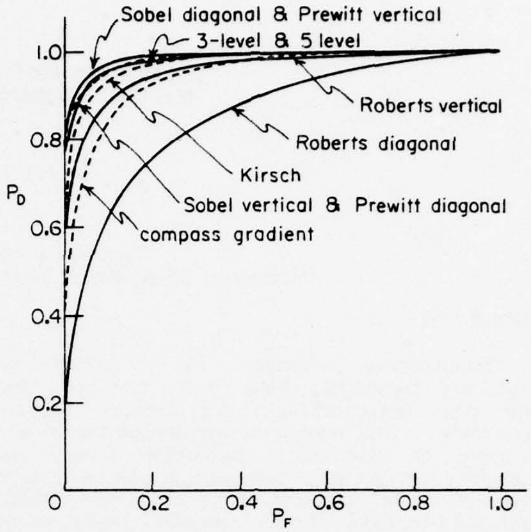


Fig. 2. Probability of detection versus probability of false detection for edge enhancement/thresholding operators.

experimentally, are useful performance indicators for edge detectors. However, these detection probability functions do not distinguish between the various types of errors that can be introduced by an edge detector. Pratt [1,p.495] has developed a simple figure of merit for edge detectors that provides a relative penalty for fragmented, smeared, and offset edges. The figure of merit measurement procedure utilizes a square array of pixels with a vertically oriented ramp edge in its center. The edge parameters and noise level can be varied to generate test edges which are then processed by an edge detector to produce binary edge maps. The figure of merit is defined as

$$F = \frac{1}{\max\{I_I, I_A\}} \sum_{i=1}^{I_A} \frac{2}{1+\alpha d^2(i)} \quad (6)$$

where I_I and I_A are the number of ideal and actual edge points, d(i) is the pixel miss distance of the i-th edge detected, and α is a scaling constant chosen to be $\alpha = 1/9$ to provide a relative penalty between smeared edges and isolated, but offset, edges. This technique can be extended to diagonal edges.

Figures 3 and 4 contain figure of merit plots as a function of signal-to- noise ratio for square root differential and template matching operators. The curves indicate that among the class of differential operators, the Prewitt and Sobel operators provide a substantially higher figure of merit than the Roberts operator. The Prewitt operator exhibits a somewhat larger figure of merit than the Sobel operator for a vertical edge, while for a diagonal edge, their performances are nearly the same. For the template operators, the 3-level, 5-level, and Kirsch operators are clearly superior to the compass gradient operator. The 3-level operator is dominant by a slight margin at all signal-to-noise ratios for diagonal edges, but for vertical edges, the relative dominance changes with signal- to-noise ratio. The Prewitt square root differential operator gives a slightly higher figure of merit than the 3-level template matching operator for vertical edges. For diagonal edges, the reverse is true.

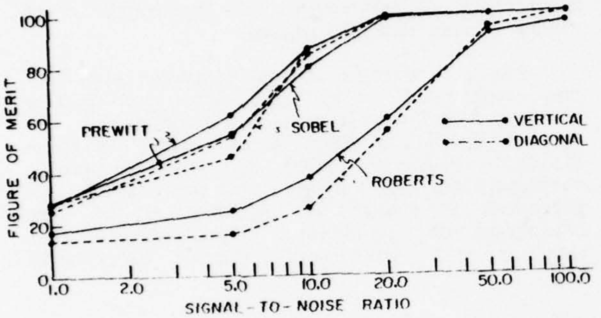


Fig. 3. Figure of merit for square root differential operators.

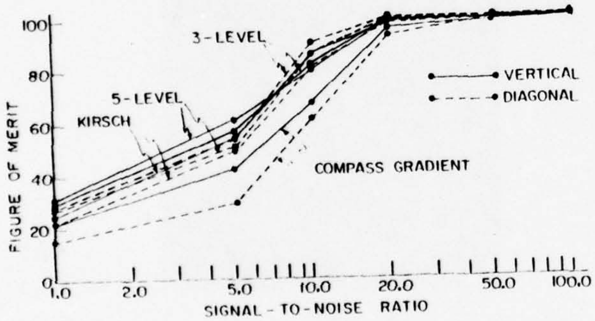


Fig. 4. Figure of merit for template matching operators.

A figure of merit comparison of the Hueckel and Abdou edge fitting operators is presented in Figure 5. The curves indicate that the Abdou operator is clearly superior to the Hueckel operator at low signal-to-noise ratio.

IMAGE TEXTURE

Image texture is a region property or feature of an image that characterizes the structural relationship of pixels within the region. The *structural relationship* of texture may be regarded from a deterministic or stochastic standpoint. In the deterministic formulation [4,5], texture is considered as a basic local pattern that is periodically or quasi-periodically repeated over some area. This definition is applicable to line patterns such as ruled line arrays, tiling patterns, etc. The stochastic formulation, adapted here, is based on a model in which a texture region is viewed as a sample of a two-dimensional stochastic process describable by is statistical parameters. This formulation is obviously applicable to the texture fields generated from random number arrays that have been so widely used in perceptual experiments [6,7]. In addition, the formulation seems well suited for natural textures consisting of isolated areas from multi-gray level images such as grass, water, forestry, etc.

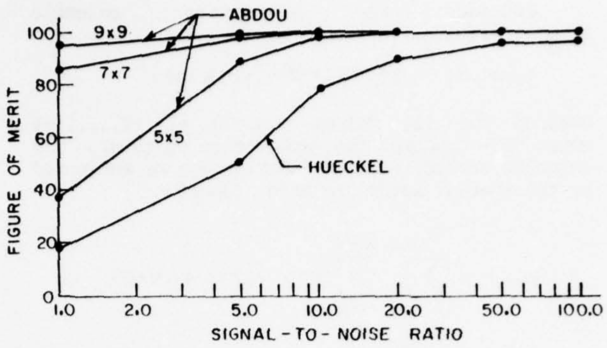


Fig. 5. Figure of merit for edge fitting operators.

STOCHASTIC TEXTURE GENERATION

Figure 6 contains a block diagram for a general model of stochastic texture generation. An array of independent, identically distributed random variables W(j,k) passes through a linear or nonlinear spatial operator $\mathcal{O}\{\cdot\}$ to produce a stochastic texture array F(j,k). By controlling the form of the generating probability density p(W) and the spatial operator, it is possible to create texture fields with specified statistical properties.

From the stochastic texture generation model of Figure 6, it is observed that fields generated by that model can be described quite compactly by specification of the spatial operator and the stationary first order probability density p(W) of the independent, identically distributed generating process W(j,k). Such information cannot generally be determined from the texture field observation F(j,k). However, this concept serves as a useful guide to the development of candidate texture features.

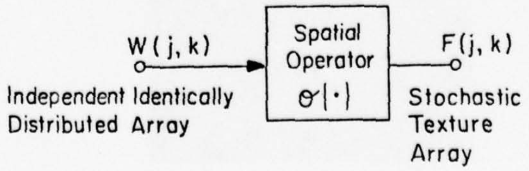


Fig. 6. Stochastic texture field operation model.

Consider the stationary ensemble autocorrelation function

$$K_F(m,n) = E\{F(j,k)F(j+m,k+n)\} \qquad (7)$$

defined for lag values $m,n = 0, \pm 1, \pm 2, \ldots, \pm T$ where $E\{\cdot\}$ denotes the expectation operator. The ensemble autocorrelation function can be estimated by the spatial autocorrelation function

$$A_F(m,n) = \sum_{u=j-W}^{j+W} \sum_{v=k-W}^{k+W} F(u,v)F(u-m,v-n) \qquad (8)$$

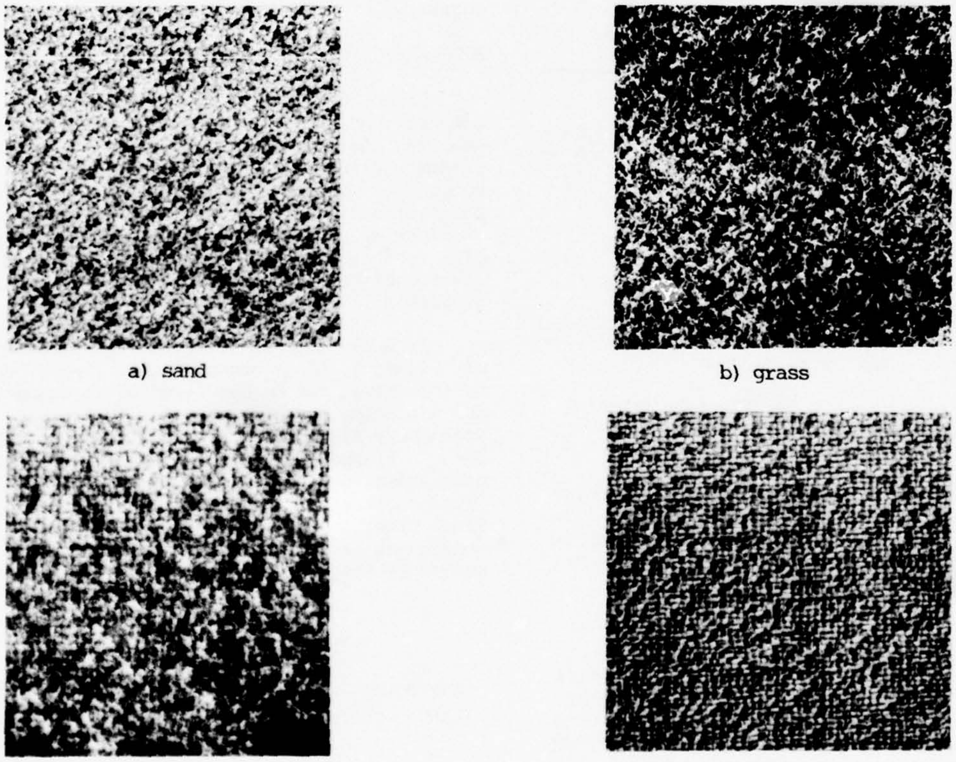
where computation is over a $(2W+1) \times (2W+1)$ window. It is possible to perform a whitening transformation, based on the measured autocorrelation function of eq. (8), to produce an uncorrelated, identically distributed field

$$\hat{W}(j,k) = H(j,k) \circledast H(j,k) \qquad (9)$$

where $H(j,k)$ is the whitening operator. The whitened field $\hat{W}(j,k)$ can be utilized as an estimate of the independent, identically distributed generating process $W(j,k)$.

If $W(j,k)$ were known exactly, then in principle, system identification techniques could be employed to estimate the spatial operator $\mathcal{O}\{\cdot\}$ from the texture observation $F(j,k)$. But, the whitened field estimate $\hat{W}(j,k)$ will only identify the spatial operator in terms of the autocorrelation function of $F(j,k)$, which is not unique. Thus, it is concluded that the probability density of the whitened field $p(\hat{W})$ and the spatial autocorrelation function of the texture field $K_F(m,n)$ are, in general, incomplete descriptors of the stochastic process $F(j,k)$. But, it may be possible that they are sufficient descriptors of its texture from the standpoint of visual texture discrimination.

Figure 7 contains several texture fields from the Brodatz [8] album that have been used as prototypes for experimentation. Examples of the measured spatial autocorrelation function of these fields are given in Figure 8. Whitened fields corresponding to these texture fields are presented in Figure 9. Examination of the histograms of the whitened fields indicates that they are all different. These experiments



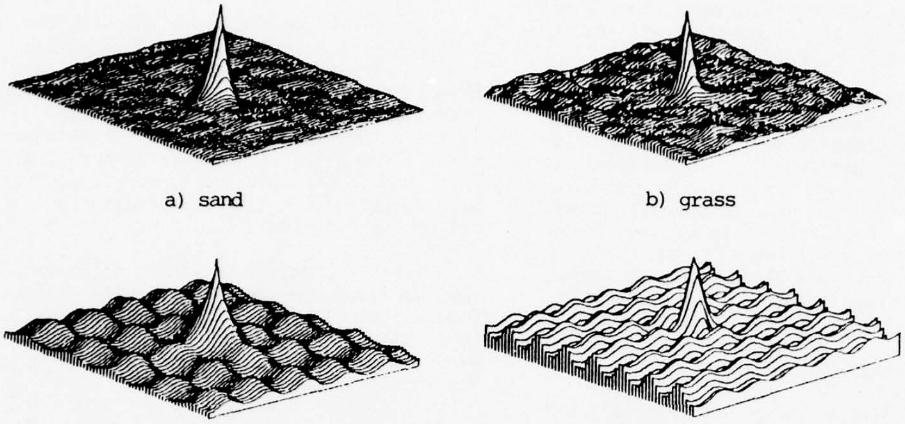
a) sand

b) grass

c) wool

d)raffia

Fig. 7. Examples of Brodatz texture fields.



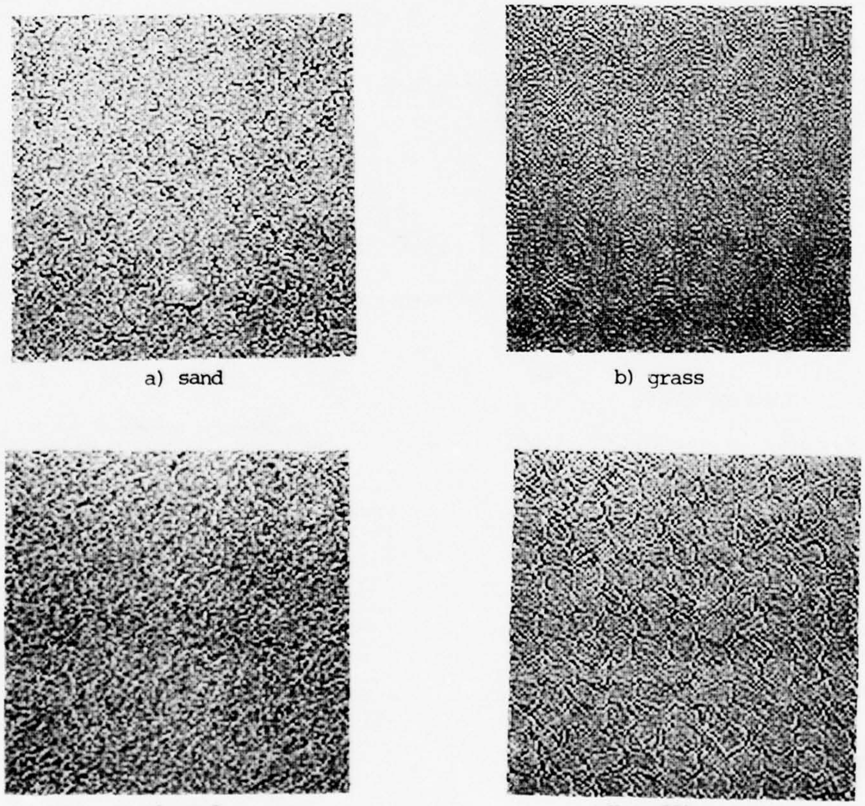
a) sand

b) grass

c) wool

d) raffia

Fig.8. Perspective views of autocorrelation functions of natural texture fields.



a) sand

b) grass

c) wool

d) raffia

Fig. 9. Whitened natural texture fields.

qualitatively support the contention that the spatial autocorrelation function of a texture field plus the first order amplitude histogram of its whitened texture field provide sufficient information for texture discrimination.

An obvious disadvantage of the whitening operator method of texture field decorrelation is the large amount of computation involved in the process. The experimental autocorrelation function of a texture block must first be formed, then the whitening operator must be generated, and finally the block must be processed. An alternative to this procedure is to utilize a gradient operator, such as a Laplacian or Sobel operator, that approximates the whitening operator.

TEXTURE FEATURE EXTRACTION

Figure 10 contains a block diagram of the stochastic-based texture feature extraction method. In the general system, the spatial autocorrelation function is measured and used to develop a decorrelation operator. A histogram of the decorrelated texture field is then measured. The texture features include moments of the histogram and spread measures of the autocorrelation function.

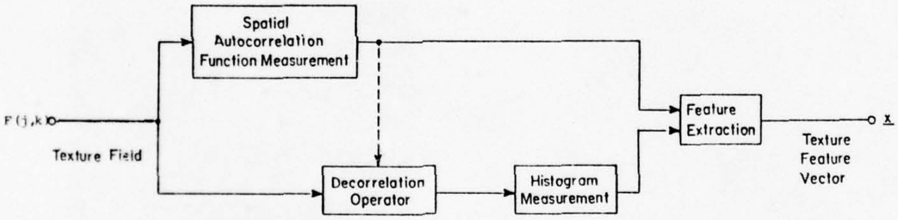


Fig. 10. Stochastic-based texture feature extraction method.

BHATTACHARYYA DISTANCE FIGURE OF MERIT

The texture features previously developed have been evaluated according to their Bhattacharyya distance [9,p.268] figure of merit for texture prototypes. The Bhattacharyya distance (B-distance for simplicity) is a scalar function of the probability densities of features of two classes defined as

$$B(S_1,S_2)=-\ln\{\int[p(\underline{x}|S_1)p(\underline{x}|S_2)]^{\frac{1}{2}}d\underline{x}\} \qquad (10)$$

where $\underline{x}$ denotes a feature vector with conditional density $p(\underline{x}|S_i)$ for class S_i. It can be shown that the B-distance is monotonically related to the Chernoff bound of the probability of classification error using a Bayes classifier. The bound on the error probability is

$$P \leq [P(S_1)P(S_2)]^{\frac{1}{2}}\exp\{-B(S_1,S_2)\} \qquad (11)$$

where $P(S_i)$ represents the a priori class probabil·ty.

The B-distance has been computed for several feature vector sets of prototype natural texture fields. In these experiments, the texture fields have been subdivided into 64 non-overlapping prototype regions of 64x64 pixels. Texture features have been extracted from each region and formed into a texture feature vector. Next, the mean and covariance of the feature vector have been computed to obtain the B-distance for pairs of prototype fields.

Table 1

Bhattacharyya Distance of Texture
Feature Sets for Prototype Texture Fields

FIELD	PAIRS	SET #1	SET #2	SET #3	SET #4
GRASS	SAND	1.15	4.39	5.64	17.78
GRASS	RAFFIA	2.10	1.15	3.33	8.51
GRASS	WOOL	0.97	1.68	2.77	16.04
SAND	RAFFIA	0.92	12.09	13.70	2.56
SAND	WOOL	1.72	11.76	13.39	9.98
RAFFIA	WOOL	2.78	4.03	7.30	7.29
AVERAGE		1.61	5.85	7.69	10.36

SET #1: 4 Autocorrelation Shape Features

SET #2: 4 Histogram Moment Features, Whitened Fields

SET #3: Combination, Whitened Fields

SET #4: 4 Histogram Moment Features, Sobel Fields

Table 1 contains a listing of B-distances for four texture feature sets. With feature set 1, four autocorrelation shape features have been used to characterize the texture field. The B-distances of the table correspond to misclassification error bounds from about 6% to 20%. These measurements indicate that autocorrelation shape features of texture fields, by themselves, are probably not adequate for texture classification. Feature set 2 consists of the first four histogram moments of the whitened texture field. The average B-distance is quite high, but some distances are small. The conclusion is that texture features based on the histogram shape of the whitened texture field may be marginally adequate. Feature set 3 combining the autocorrelation and whitened histogram features provides a large average B-distance and also a large minimum distance. The worst case is a 3.1% classification error bound. Feature set 4 has yielded remarkable performance. The B-distances obtained for four histogram moments and no autocorrelation shape information using a Sobel operator for decorrelation are extremely large on average. Again, the worst case represents a misclassification error bound of about 3%. This result is extremely encouraging since the Sobel operator and the histogram measurement can be implemented by smart sensor signal processing.

AKNOWLEDGEMENT

The work on edge detection reported here is abstracted from a joint study with Dr. Ikram Abdou, now a research scientist at IBM in San Jose, California. The research on texture represents a collaborative effort with Dr. O.D. Faugeras who is with the Institut de Recherche d'Informatique et d'Automatique, Domaine de Voluceau-Rocquencourt, 78150 Le Chesnay, France.

REFERENCES

1. W.K. Pratt, Digital Image Processing, Wiley-Interscience, New York, 1978.

2. M.H. Hueckel, "An Operator Which Locates Edges in Digitized Pictures," J. Assoc. Comput. Mach., Vol.18, No.1, January 1971, pp. 113-125.

3. I. Abdou, "Quantitative Methods of Edge Detection," University of Southern California, Image Processing Institute, USCIPI Report 800, Los Angeles, California, 1978.

4. R.M. Pickett, "Visual Analysis of Texture in the Detection and Recognition of Objects," in Picture Processing and Psychopictorics, B.S. Lipkin and A. Rosenfeld, Eds., Academic Press, New York, pp. 289-308, 1970.

5. J.K. Hawkins, "Texture Properties for Pattern Recognition," in Picture Processing and Psychopictorics, B.S. Lipkin and A. Rosenfeld, Eds., Academic Press, New York, pp. 347-370, 1970.

6. B. Julesz, "Visual Pattern Discrimination," IRE Transactions Information Theory, Vol. IT-80, No. 1, pp. 84-92, February 1962.

7. W.K. Pratt, O.D. Faugeras, and A. Gagalowicz, "Visual Discrimination of Stochastic Texture Fields," IEEE Transactions on Systems, Man, and Cybernetics, November 1978.

8. P. Brodatz, Texture: A Photograph Album for Artists and Designers, Dover, New York, 1956.

9. K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, New York, 1972.

SOME EXPERIMENTS IN MATCHING USING RELAXATION

Azriel Rosenfeld

Computer Vision Laboratory
Computer Science Center
University of Maryland
College Park, MD 20742

ABSTRACT

This paper describes several experiments in point pattern matching and graph matching. Point patterns can be correlated by counting, for each relative position, the number of pairs that lie sufficiently close together. A more flexible approach is to use relaxation to assign confidences to pairings of the points, based on local pattern matches. Relaxation can also be used to match graphs, yielding sets of pairings with associated confidences.

1. Point pattern matching [1]

Let $\mathcal{P} \equiv P_1,\ldots,P_m$ and $\mathcal{Q} \equiv Q_1,\ldots,Q_n$ be two point patterns. For each pair (P_i,Q_j), if we shift the patterns so that P_i and Q_j coincide, other pairs of points (P_h,Q_k) may also coincide within some tolerance. The number of such pairs is a measure of how well the patterns match under that particular shift. If $\mathcal{P}$ and $\mathcal{Q}$ have many points in common, the shift that maps these points into themselves will receive a high score, while other shifts will receive at best low scores resulting from accidental correspondences between a few pairs of points.

Figures 1-4 show two examples of this simple matching process, which is related to cross-correlating one pattern with a "blurred" version of the other (in which each point has been expanded into a disk). In these examples, the tolerance was taken to be 5% of the smaller interpoint distance, i.e., of $\min[P_iP_h,Q_jQ_k]$. The smearing and "echoes" in Figure 4 are due to matches of the edges of the tank with each other or with other parts of themselves; this does not occur when isolated feature points, rather than edge points, are used, as in Figure 2. Nevertheless, the correct match peak is higher by nearly an order of magnitude than the echoes.

Further details on this matching scheme, and additional examples, can be found in [1], which also studies the sensitivity of the process to various types of noise and distortion, including random displacements of the points and rotation or rescaling of one pattern relative to the other.

2. Point pattern matching by relaxation [2]

An alternative approach to point pattern matching is to consider all possible pairings of the points, and eliminate (or reduce the confidence of) those pairings that define displacements for which the points match poorly. Specifically, consider the pairing of P_i with Q_j; let $\underline{\delta}_{ij}(h,k)$ be the position difference between P_h and Q_k when P_i is matched with Q_j; and let the support given to (P_i,Q_j) by a pair of points having position difference $\underline{\delta}$ be $\phi(\underline{\delta})$ (e.g., one might use $\phi(\underline{\delta}) = \frac{1}{1+|\delta|^2}$, say). Then we can define an estimate of our current confidence in (P_i,Q_j) as, for example,

$$c^{(r+1)}(P_i,Q_j)=\frac{1}{m-1}\sum_{h\neq i}\max_{k\neq j}\{\min\,[\phi(\underline{\delta}_{ij}(h,k)),c^{(r)}(P_h,Q_k)]\}$$

where $c^{(0)}(P_i,Q_j) = 1$. When this process of support computation is iterated, the confidences $c(P_i,Q_j)$ of pairings that correspond to a good match remain relatively high, while those of other pairings become very low. Two examples, corresponding to Figures 2 and 4, are shown in Figures 5-6. A more detailed discussion and further examples can be found in [2].

3. Discrete graph matching [3]

Point pattern matching provides a possible approach to distortion-tolerant image matching based on patterns of feature points extracted from the image. On a more abstract level, one may be interested in matching image descriptions, rather than the images themselves. Such a description often takes the form of a labelled graph, in which the nodes correspond to regions or local features in the image; the node labels are region descriptors; and the arcs represent relationships (not necessarily spatial) between pairs of nodes. In this section we discuss the case where the labels are symbolic, and we are looking for exact matches; in Section 4 we will consider the case where the labels are numerical and the matching is quantitative.

Let G and H be two labelled connected graphs, and suppose we want to find labelled subgraphs of G that are isomorphic to H. Initially, we assume that any node of G, say with label λ, can be any one of the nodes of H that have label λ. For any such pairing, say of node n with node m, let the neighbors of m in H be $(m_1,\ldots,m_u)$, and let the neighbors of n in G be $(n_1,\ldots,n_v)$. We now check that for each m_i there exists at least one n_j for which (m_i,n_j) is still a possible pair; if not, we discard the pair (m,n). This process is iterated until no further discards occur.

If the graph structure and labels of G are very ambiguous, the process just described may result in a high degree of residual ambiguity. In many cases, however, the results become quite unambiguous after only a few iterations. For example, let G be the adjacency graph of the 48 contiguous states in the U.S., and let the nodes of G be labelled with the first letters of the names of the state capitals. We can define H's by randomly selecting connected subgraphs of G having given numbers of nodes. Figure 7 shows the results of applying the process described in the preceding paragraph, using a set of such H's. We see that the process stabilizes after about three iterations, and leads to very low ambiguity (as measured by the average number of G nodes that are still paired with each H node).

4. Weighted graph matching [4]

Suppose now that the graph labels are numerical rather than symbolic; we now want to find subgraphs of G that are isomorphic to H and for which the label values match closely. More precisely, we will compute a confidence for every possible pairing of a node m of H with a node n of G; we want this confidence to be high when we have an isomorphism that makes m and n correspond and that has good value matches, and low otherwise.

The confidences c(m,n) will be computed by a relaxation process analogous to that described in Section 2. We will assume that the arcs, as well as the nodes, have numerical labels; as an example, consider a graph of highway connections between cities, where the cities are labelled with their populations, and the intercity connections are labelled with their mileages. Initially, we get $c^{(0)}(m,n) = \phi(\delta)$, where δ is the discrepancy between the values at m and n, as in Section 2. Let the neighbors of m be $m_1,\ldots,m_u$, and those of n be $n_1,\ldots,n_u$. For any pair (m_i,n_j), let $\delta_{mn}(i,j)$ be the discrepancy between the values on arcs (m,m_i) and (n,n_j). Then we can compute a new estimate of c(m,n) as, for example,

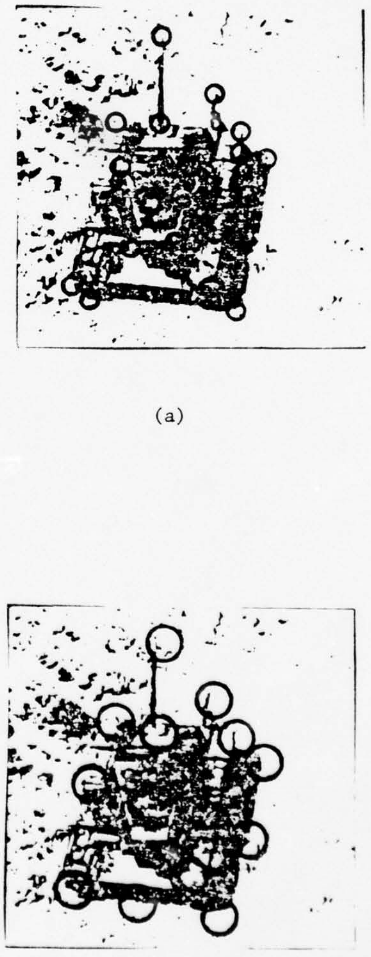
$$c^{(r+1)}(m,n) =$$

$$\min_{m_i} \max_{n_j} \{\min[\phi(\delta_{mn}(i,j)),c^{(r)}(m,n),c^{(r)}(m_i,n_j)]\}$$

An example of this matching technique, using an intercity mileage graph, is shown in Figure 8 The subgraphs H were randomly chosen, and the values of their node and arc labels were modified by adding random noise to them. After a few iterations, the confidence values tended to stabilize, and the match merit, as defined by the average confidence of the correct pairs, minus that of the incorrect pairs, was quite high.

REFERENCES

1. D. J. Kahl, A. Rosenfeld, and A. Danker, Some experiments in point pattern matching, University of Maryland, Computer Science Center Technical Report 690, September 1978.
2. S. Ranade and A. Rosenfeld, Point pattern matching by relaxation, University of Maryland, Computer Science Center Technical Report 702, October 1978.
3. L. Kitchen, Discrete relaxation for matching relational structures, University of Maryland, Computer Science Center Technical Report 665, June 1978.
4. L. Kitchen and A. Rosenfeld, Weighted graph matching by relaxation, University of Maryland, Computer Science Center Technical Report, in preparation.



(a)

(b)

Figure 1

Two sets of feature points in a picture of a tank, chosen independently by two people.

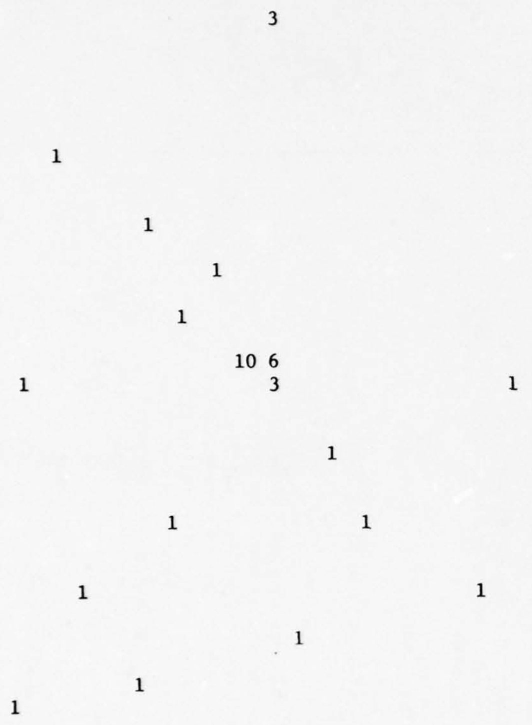


Figure 2

Array of match scores for various relative displacements of the two point patterns shown in Figure 1.

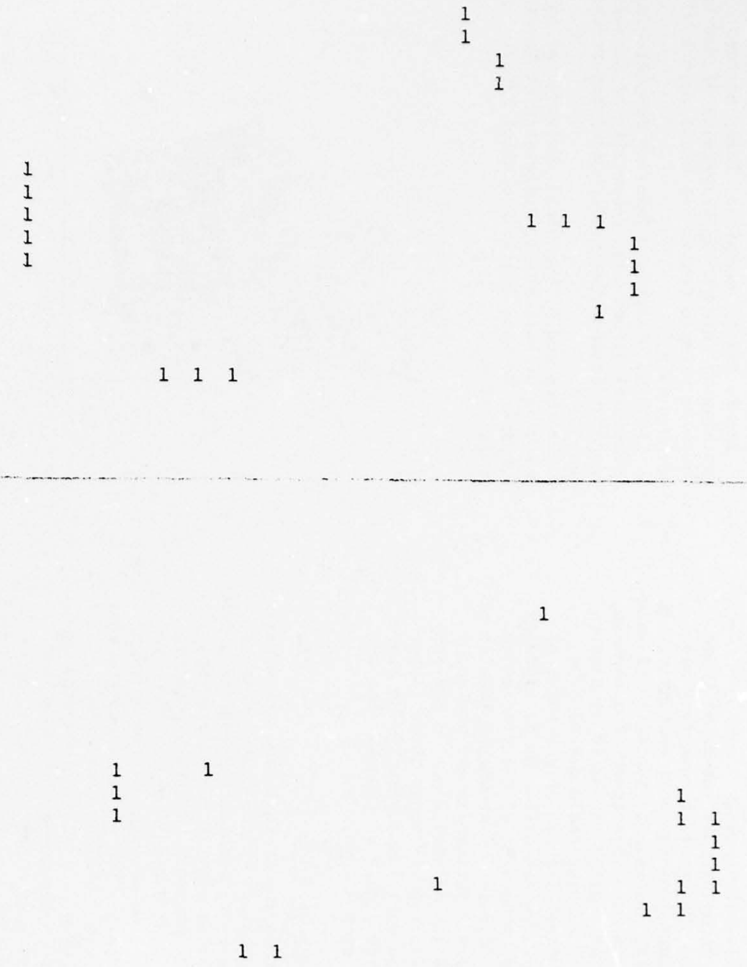


Figure 3

Two sets of thinned edge detector responses in a FLIR image of a tank, obtained using two different edge detectors.

```
                                            1  1  1
                                                                                            1
                                                                                         1  3

                                                        1
  1                                                     1                                 1  1
  3                                                  3  3                              1  1
  1                                                  3 10                                             1
                                                     1 45                                             1
                                                     1 37     1  1                                       1
                                                     1  6                                             1  3
                 1  2                                1  1                                             3  6
              1  1                                         1     1                                    3  6
              2  1                                                                                    1  3
              5  2                                                                                    1  1
              1  3  1  2  1                                      1                                    1
                 1                                                  3
                 1                                                  6
                                                                    3  1
                                                                 1     1
                                                                       1
                          3                                         1
                          2
                             2  1
                             1
```

Figure 4

Array of match scores for the point patterns of Figure 3.

```
                       1








                           1        3  1
                  1        6        1
                                    3
                              7 13
        1        8           8        2
                       23 53            1
  1        6       32 11        6        1
        1        7  9

     1        3  8  4  5
           2     3
           2
              3              1

                                    1
```

Figure 5

Array of match scores for the point patterns of Figure 1 obtained by relaxation. For each displacement, the sum of the confidences (x100) for all point pairs having that displacement is displayed.

```
                       1
                       2   1

                                                                           8
                                                           1   5      13  27  12
       1                                                       8      19  66  61
   1   2                                                      11      26 147  80
   2   3                                                      13  22  88 347  91  22
   3   3                                                      25  42  59 281  55
   4   4                                                      11  17  25 112  49  17
   4   3                                                       9  13  18  38  34  11
   3   1                                                       7  17  21  10   9   6
   2   1                                               4   5       5  13  14   5
       1                                               3   4   4
                                                       2   6   8
                                                       3  11   3
                                                       6  15
                                                       5   7   2
                                                       5   2   4   1
                                                       4   2   1   1
                                                       1   2   1
```

Figure 6

Analogous to Figure 5, for the point patterns of Figure 3.

Nodes in H	Iterations to stabilize	No. of matchings (x000)	No. of excess pairs
4	3.4	4.8	0.4
5	3.6	5.8	1.0
6	3.6	7.7	0.2
7	3.6	8.6	0.4
8	3.6	12.1	0.6
9	3.8	13.3	0.0
10	4.2	14.3	0.4
11	3.8	16.8	0.8
12	4.0	19.1	0.6
13	4.0	21.8	0.2

Figure 7

Results of a set of exact graph matching experiments. Graph G represented the adjacencies of the 48 contiguous states of the U.S. Each case is an average of five randomly constructed examples.

	Iteration No.				
Noise range (%)	1	2	3	4	5
0	0.98	1.00	1.00	1.00	1.00
±10	0.79	0.95	1.00	1.00	1.00
±20	0.57	0.78	0.94	1.00	1.00
±30	0.33	0.72	0.96	0.99	0.99
±40	0.32	0.73	0.92	0.99	0.99
±50	0.23	0.83	0.98	0.99	0.99
±100	0.10	0.53	0.86	0.86	0.90

Figure 8

Results of a set of weighted graph matching experiments. Graph G represented intercity highway mileage and population data for 44 Eastern U.S. cities. Each case is an average of four randomly constructed examples. The noise was uniformly distributed in the indicated interval. The entries are values of $\overline{T}-\overline{F}$, where $\overline{T}$ is the average weight of the correct pairings, and $\overline{F}$ the average weight of the incorrect pairings, after rescaling to make the highest weight 1; thus $-1 \leq (\overline{T}-\overline{F}) \leq 1$.

CALCULATING THE REFLECTANCE MAP

Berthold K. P. Horn
Robert W. Sjoberg

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT

It appears that the development of machine vision may benefit from a detailed understanding of the imaging process. The reflectance map, showing scene radiance as a function of surface gradient, has proved to be helpful in this endeavor. The reflectance map depends both on the nature of the surface layers of the objects being imaged and the distribution of light sources. Recently, a unified approach to the specification of surface reflectance in terms of both incident and reflected beam geometry has been proposed. The reflecting properties of a surface are specified in terms of the bidirectional reflectance-distribution function (BRDF).

Here we derive the reflectance map in terms of the BRDF and the distribution of source radiance. A number of special cases of practical importance are developed in detail. The significance of this approach to the understanding of image formation is briefly indicated.

1. THE REFLECTANCE MAP

The apparent "brightness" of a surface patch depends on the orientation of the patch relative to the viewer and the light sources. Different surface elements of a non-planar object will reflect different amounts of light towards an observer as a consequence of their differing attitude in space. A smooth opaque object will thus give rise to a "shaded" image, one in which "brightness" varies spatially, even though the object may be illuminated evenly and covered by a uniform surface layer. This shading provides important information about the object's shape and has been exploited in machine vision [1 - 8].

A convenient representation for the relevant information is the "reflectance map" [4, 6]. The reflectance map, R(p, q), gives scene radiance as a function of surface gradient (p, q) in a viewer centered coordinate system. If z is the elevation of the surface above a reference plane lying perpendicular to the optical axis of the imaging system, while x and y are distances in this plane measured parallel to orthogonal coordinate axes in the image, then p and q are the first partial derivatives of z with respect to x and y respectively.

$$p = \partial z/\partial x \quad \text{and} \quad q = \partial z/\partial y$$

The reflectance map is usually depicted as a series of contours of constant scene radiance (Fig. 1). It can be measured experimentally using a goniometer-mounted sample or an image of an object of known shape. Alternatively, a reflectance map may be calculated if properties of the surface material and the distribution of light sources are given. One purpose of this paper is to provide a systematic approach to this latter endeavor. Another is to derive the relationship between scene radiance and image irradiance in an imaging system. This is relevant to machine vision since "gray-levels" are quantized measurements of image irradiance.

2. MICRO-STRUCTURE OF SURFACES

When a ray of light strikes the surface of an object it may be absorbed, transmitted or reflected. If the surface is flat and the underlying material homogeneous, the reflected ray will lie in the plane formed by the incident ray and the surface normal and will make an angle with the local normal equal to the angle between the incident ray and the local normal. This is referred to as "specular", "metallic" or "dielectic" reflection. Objects with surfaces of this kind form virtual images of surrounding objects.

Many surfaces are not perfectly flat on a microscopic scale and thus "scatter" parallel incident rays into a variety of directions (Fig. 2a). If deviations of the local surface normals from the average are small, most of the rays will lie near the direction for ideal specular reflection and contribute to a surface "shine" or "gloss".

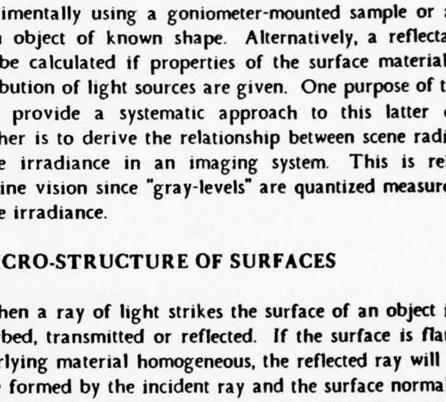


Figure 1: A typical reflectance map for a surface, with both a glossy and a matte component of reflection, illuminated by a point source. The coordinates are surface slope in the x and y directions, and the curves shown are contours of constant scene radiance.

Other surface layers are not homogeneous on a microscopic scale and thus "scatter" light rays which penetrate the surface by refraction and reflection at boundaries between regions with differing refractive indeces (Fig. 2b). Scattered rays may re-emerge near the point of entry with a variety of directions and so contribute to "diffuse", "flat" or "matte" reflection. Snow and white paint layers are examples of surfaces with this kind of behavior. Frequently both effects occur in surface layers, with some rays reflected at the nearly flat outer surface of the object, while others penetrate deeper and re-emerge after multiple refractions and reflections in the inhomogeneous interior.

In each case, the distribution of reflected light depends on the direction of incident rays and the details of the microstructure of the surface layer. Naturally, what constitutes microstructure depends on one's point of view. Usually surface structures not resolved in a particular imaging situation are taken here to be microstructure. When viewing the moon through a telescope for example, smaller "hillocks" and "craterlets" are part of this microstructure. This consideration leads to more complicated models of interaction of iight with surfaces than those discussed so far. It is possible, for example, to consider an undulating surface covered with a material which in itself already has complicated reflecting behavior (Fig. 2c).

Reflectance is not altered by rotating a surface patch about its normal when there is no asymmetry or preferred direction to either the pattern of surface undulations or the distribution of sub-surface inhomogeneities. Many surface layers behave this way and permit a certain degree of simplification of the analysis. Exceptions are such things as diffraction gratings, iridescent plumage and the mineral called "tiger eye". These all have a distinct directionality in their surface micro-structure and will not be considered here any further.

Considerable attention has been paid to the reflective properties of various surface layers. A few researchers have concentrated on the experimental determination of surface reflectance properties [9 - 21]. At the same time, many models have been developed for surface layers based on some of the considerations presented above [22 - 35]. Models often are too simple to be realistic, or too complicated to yield solutions in closed form. In the latter case, Monte-Carlo methods can be helpful, although they only lead to numerical specification of the reflecting behavior. Purely phenomenological models of reflectance have found favor in the computer graphics community [36, 37, 38]. Several books have appeared describing the uses of reflectance measurements in determining basic optical properties of the materials involved [39, 40, 41]. Attention has been paid, too, to the problem of making precise the definitions of reflectance and related concepts [42, 43].

3. RADIOMETRY

A modern, precise nomenclature for radiometric terms has been promoted by a recent NBS publication [43]. The following short table gives the terms, preferred symbols and unit dimensions of the radiometric concepts we will have occasion to use for the development presented here:

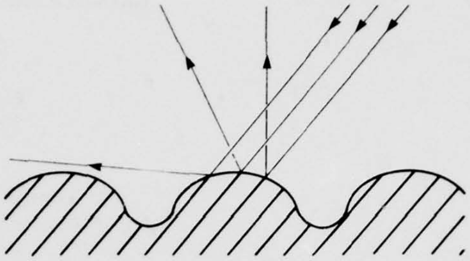


FIGURE 2a: Undulations in a specularly reflecting surface causing scattering of incident rays into a variety of directions. The surface will not appear specular if it is imaged on a scale where the surface undulations are not resolved. It may instead have a glossy appearance.

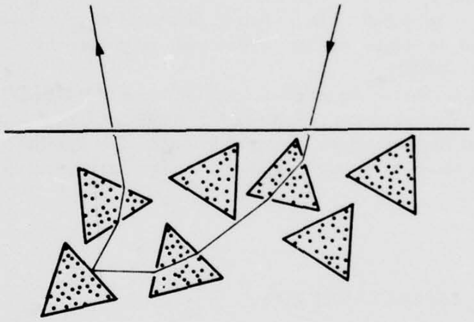


FIGURE 2b: Inhomogeneities in refractive index of surface layer components causecident rays to be scattered into a variety of directions upon reflection. This kind of surface micro-structure gives rise to matte reflection.

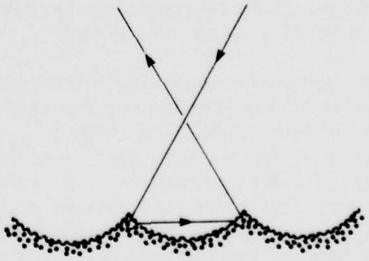


FIGURE 2c: Compound surface illustrating more complex model of interaction of light rays with surface microstructure.

RADIOMETRIC CONCEPTS

Radiant flux	Φ	(W)
Radiant Intensity	$I = d\Phi/d\omega$	$(W \,.\, sr^{-1})$
Irradiance	$E = d\Phi/dA$	$(W \,.\, m^{-2})$
Radiant Exitance	$M = d\Phi/dA$	$(W \,.\, m^{-2})$
Radiance	$L = d^2\Phi \,/\, (dA \,.\, \cos\theta \,.\, d\omega)$	$(W \,.\, m^{-2} \,.\, sr^{-1})$

Radiant flux, Φ, is the power propagated as optical electromagnetic radiation and is measured in watts (W). The radiant intensity, **I**, of a source is the exitant flux per unit solid angle and is measured in watts per steradian $(W \,.\, sr^{-1})$. The total flux emitted by a source is the integral of radiant intensity over the full sphere of possible directions (4π steradians). The irradiance, E, is the incident flux density, while radiant exitance, M, is the exitant flux density, both measured in watts per square meter of surface $(W \,.\, m^{-2})$. The total radiant exitance equals the total irradiance if the surface reflects all incident light, absorbing and transmitting none.

The radiance, **L**, is the flux emitted per unit *foreshortened* surface area per unit solid angle. Radiance is measured in watts per square meter per steradian $(W \,.\, m^{-2} \,.\, sr^{-1})$. It can equivalently be defined as the flux emitted per unit surface area per unit *projected* solid angle. Radiance is an important concept since the apparent "brightness" of a surface patch is related to its radiance. Specifically, image irradiance will be shown to be proportional to scene radiance.

Radiance is a directional quantity. If the angle between the surface normal and the direction of exitant radiation is θ, then the term "foreshortened area" stands for the actual surface area times the cosine of this angle θ. Similarly the "projected solid angle" stands for the actual solid angle times the cosine of the angle θ. Here we will use the symbol ω to denote a solid angle, while Ω will be used to denote a *projected* solid angle. If $d\omega$ and $d\Omega$ are corresponding infinitesimal solid angles and projected solid angles respectively, then

$$d\Omega = d\omega \,.\, \cos\theta$$

The following example (Fig. 3) will illustrate some of these ideas. Consider a source of radiation with intensity **I** in the direction of a surface patch of area **dA**, oriented with its surface normal making angle θ with the line connecting the patch to the source. In fact, as seen from the source, it appears only as large as a patch of area **dA** $\cos\theta$ oriented perpendicular to this line. The corresponding solid angle is simply the area of this equivalent patch divided by the square of the distance from the source to the patch. Thus,

$$d\omega = dA \cos\theta \,/\, r^2 \qquad (sr)$$

The flux intercepted then is

$$d\Phi = I\, d\omega = I\, dA \cos\theta \,/\, r^2 \qquad (W)$$

The irradiance of the surface is just the incident flux divided by the area of the surface patch.

$$E = d\Phi/dA = I \cos\theta \,/\, r^2 \qquad (W \,.\, m^{-2})$$

4. THE BIDIRECTIONAL REFLECTANCE-DISTRIBUTION FUNCTION

The Bidirectional Reflectance-Distribution Function (BRDF) was recently introduced by Nicodemus, Richmond, Hsia, Ginsberg and Limperis [43] as a unified notation for the specification of reflectance in terms of both incident- and reflected- beam geometry. The BRDF is denoted by the symbol f_r and captures the information about how "bright" a surface will appear viewed from a given direction, when it is illuminated from another given direction. To be more precise, it is the ratio of reflected radiance dL_r in the direction towards the viewer to the irradiance dE_i in the direction towards a portion of the source. In symbols,

$$f_r(\theta_i, \phi_i; \theta_r, \phi_r) = dL_r(\theta_i, \phi_i; \theta_r, \phi_r; E_i) \,/\, dE_i(\theta_i, \phi_i) \qquad (sr^{-1})$$

Here, θ and ϕ together indicate a direction, the subscript **i** denoting quantities associated with incident radiant flux, while the subscript **r** indicates quantities associated with reflected radiant flux [43].

The geometry is as depicted in the figure (Fig. 4). A surface-specific coordinate system is erected with one axis along the local normal to the surface and another defining an arbitrary reference direction in the local tangent plane. Directions are specified by polar angle θ (colatitude) measured from the local normal and azimuth angle ϕ (longitude) measured clockwise from the reference direction in the surface. In general, incident flux may arrive from many portions of extended sources, so incident radiance $L_i(\theta_i, \phi_i)$ is a function of direction. If we consider the component of flux $d\Phi_i$ arriving on the surface patch of area **dA** from an infinitesimal solid angle $d\omega_i$ in the direction (θ_i, ϕ_i) we obtain

$$d\Phi_i = L_i \cos\theta_i\, d\omega_i\, dA = dE_i\, dA$$

where $dE_i = L_i \cos\theta_i\, d\omega_i$ is the incident irradiance contributed by the portion of the source found in the solid angle $d\omega_i$ in the direction (θ_i, ϕ_i). Similarly, it is easy to see that the radiant flux emitted into an infinitesimal solid angle $d\omega_r$ in the direction (θ_r, ϕ_r) equals

$$d\Phi_r = dL_r\, d\omega_r\, dA$$

where $dL_r(\theta_r, \phi_r)$ is the radiance in the direction (θ_r, ϕ_r) due to the reflection of the incident flux. The BRDF is then defined as follows:

$$f_r(\theta_i, \phi_i; \theta_r, \phi_r) = (d\Phi_r/d\omega_r) \,/\, d\Phi_i = dL_r/dE_i$$

and thus has dimension inverse steradian (sr^{-1}). The BRDF allows one to obtain reflectance for any defined incident and reflected ray geometry simply by integrating over the specified solid angles [43].

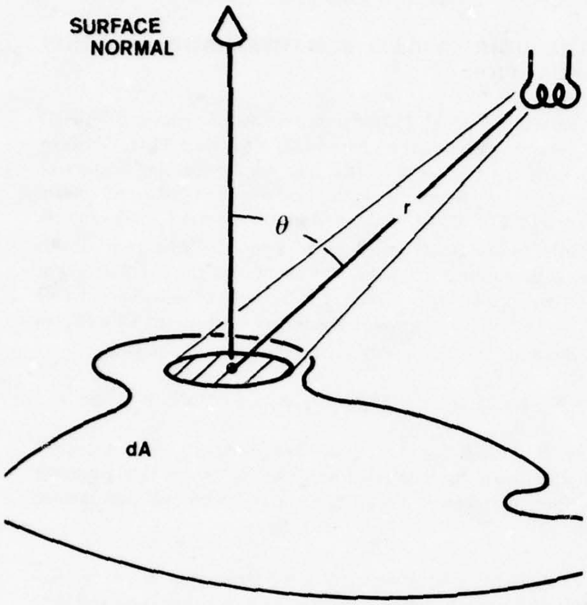


FIGURE 3: Point source illuminating a surface, illustrating basic radiometric concepts.

5. INTEGRALS OVER SOLID ANGLES AND PROJECTED SOLID ANGLES

The admitting aperture of an imaging system may occupy a significant solid angle when seen from the point of view of the objects being imaged. We will furthermore have to deal with extended sources. In both cases it is necessary to integrate various quantities over solid angles or projected solid angles. This can be accomplished by double-integration with respect to the polar and azimuth angles (Fig. 5). If X is the quantity to be integrated, we have

$$\int X\, d\omega = \int\int X \sin\theta\, d\theta\, d\phi$$

and

$$\int X\, d\Omega = \int\int X \cos\theta \sin\theta\, d\theta\, d\phi$$

If for example X = 1 and the region of integration is the hemisphere above the object's surface, then

$$\int X\, d\omega = \int_{-\pi}^{\pi} \int_{0}^{\pi/2} \sin\theta\, d\theta\, d\phi = 2\pi$$

while

$$\int X\, d\Omega = \int_{-\pi}^{\pi} \int_{0}^{\pi/2} (1/2) \sin 2\theta\, d\theta\, d\phi = \pi$$

The latter result will be used in the discussion of perfectly diffuse reflectance.

6. PERFECTLY DIFFUSE REFLECTANCE

A perfectly diffuse or "lambertian" surface appears equally "bright" from all directions, regardless of how it is irradiated, and reflects all incident light [43]. Thus the reflected radiance is isotropic, that is L_r is constant, with the same value for all directions (θ_r, ϕ_r). Also the integral of reflected radiance over the hemisphere above the surface must equal the irradiance E_i. This implies that the BRDF for this ideal surface. $f_{r,id}$ is constant, and that the radiant exitance, M, equals the irradiance E. If the reflected radiance is L_r, then the radiant exitance can be found by integration.

$$M = \int L_r\, d\Omega_r = L_r\, \pi$$

As a result one finds that

$$f_{r,id} = L_r / E_i = 1/\pi$$

If we have an extended source with radiance L_i, then the irradiance on the surface due to a small portion of solid angle $d\omega_i$ lying in the direction (θ_i, ϕ_i) is $dE_i = L_i \cos\theta_i\, d\omega_i$. So the reflected radiance is,

$$L_r = (1/\pi) \int L_i \cos\theta_i\, d\omega_i$$

This is a form of Lambert's cosine law.

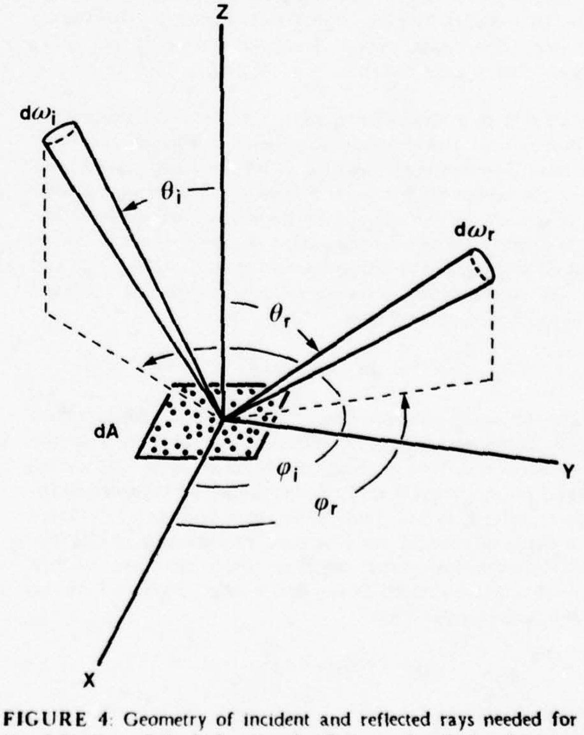


FIGURE 4: Geometry of incident and reflected rays needed for the definition of the bidirectional reflectance-distribution function (BRDF). Redrawn from [43].

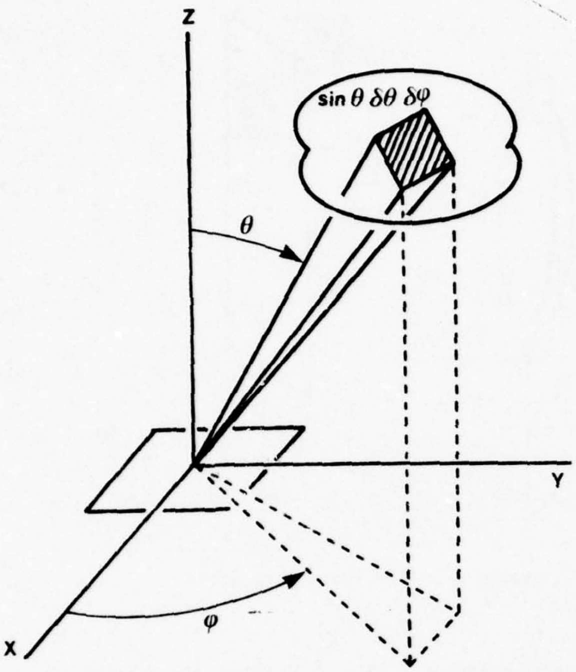


FIGURE 5: Polar and azimuth angles used in double integrals over specified solid angles.

7. COLLIMATED SOURCES AND THE DIRAC DELTA-FUNCTION

Not all sources are extended. One way to deal with sources that are highly collimated is to treat them as limiting cases of extended sources, with the distribution tending towards an impulse or delta-function. If this is to be expressed in a coordinate system of polar and azimuth angles, one has to take into account the non-uniform spacing of coordinates. Consider a collimated source which produces an irradiance E_0 on a surface oriented orthogonally to the direction (θ_0, ϕ_0) of its rays. Clearly the radiance L_i of this source should be zero except for this direction. The product of Dirac delta functions, $\delta(\theta_i - \theta_0)\ \delta(\phi_i - \phi_0)$, will be a useful ingredient of the formula expressing L_i as a function of the angles. One must, however, insure that the irradiancę on a surface lying orthogonal to the rays equals E_0.

$$E_0 = \int_{-\pi}^{\pi} \int_{0}^{\pi/2} L_i \sin\theta_i \, d\theta_i \, d\phi_i .$$

Clearly this can be accomplished if

$$L_i = E_0\ \delta(\theta_i - \theta_0)\ \delta(\phi_i - \phi_0) / \sin\theta_0$$

This is called the "double-delta" representation of source radiance for a collimated source. It can also be written in an alternate form using the identity

$$\delta[f(x) - f(x_0)] = \delta(x - x_0) / f'(x_0).$$

where $f'(x_0)$ is the derivative of $f(x)$ evaluated at $x = x_0$. Then,

$$L_i = E_0\ \delta(\cos\theta_i - \cos\theta_0)\ \delta(\phi_i - \phi_0).$$

8. PERFECTLY SPECULAR REFLECTANCE

A perfectly specular or "mirror-like" surface reflects light rays in such a way that the exitant angle ϑ_r equals the incident angle θ_i and that the incident and reflected ray lie in a plane containing the surface normal. The reflected radiance of a surface patch in the direction (θ_r, ϕ_r) is simply the source radiance in the corresponding reflected direction. That is,

$$L_r(\theta_r, \phi_r) = L_i(\theta_r, \phi_r + \pi)$$

The surface thus forms a virtual image of the source. From the definition of the BRDF, we see that

$$L_r = \int f_r \, dE_i = \int f_r \, L_i \, d\,\Omega_i$$

That is,

$$L_r = \int_{\pi}^{\pi} \int_{0}^{\pi/2} f_r \, L_i \cos\theta_i \sin\theta_i \, d\theta_i \, d\,\phi_i$$

We can satisfy the conditions stated above if we let

$$f_{r,is} = \delta(\theta_i - \theta_r)\ \delta(\phi_i - \phi_r + \pi) / (\sin\theta_i \cos\theta_i)$$

This is called the "double-delta" form of the BRDF for perfectly specular reflectance. Using the identity mentioned in the last section, we can write this in an alternate form [43]:

$$f_{r,is} = 2\ \delta(\sin^2\theta_r - \sin^2\theta_i)\ \delta(\phi_r - \phi_i + \pi)$$

9. ANALYSIS OF IMAGE FORMING SYSTEM

We will now analyze a simple image-forming system (Fig. 6). We assume that the device is properly focused; that is, those rays originating from a particular point on the object which pass through the entrance aperture are deflected to meet at a single point in the image plane. Similarly, rays originating in the infinitesimal area dA_o on the objects surface are projected into some area dA_p in the image plane and no rays from other portions of the object's surface will reach this area of the image. Further, we assume that there is no "vignetting", that is, the entrance aperture is a constant circle of diameter d and does not become smaller for directions which make a larger angle with the optical axis. The effect of vignetting on image irradiance will be considered later.

The exposure of film in a camera is proportional to image irradiance, E_p, and gray-levels in a digital imaging system are quantized measurements of image irradiance. In order to calculate image irradiance we must first determine the flux $d\Phi_L$ passing through the entrance aperture arriving from the patch of area dA_o on the object.

$$d\Phi_L = dA_o \int L_r \, d\Omega_r$$

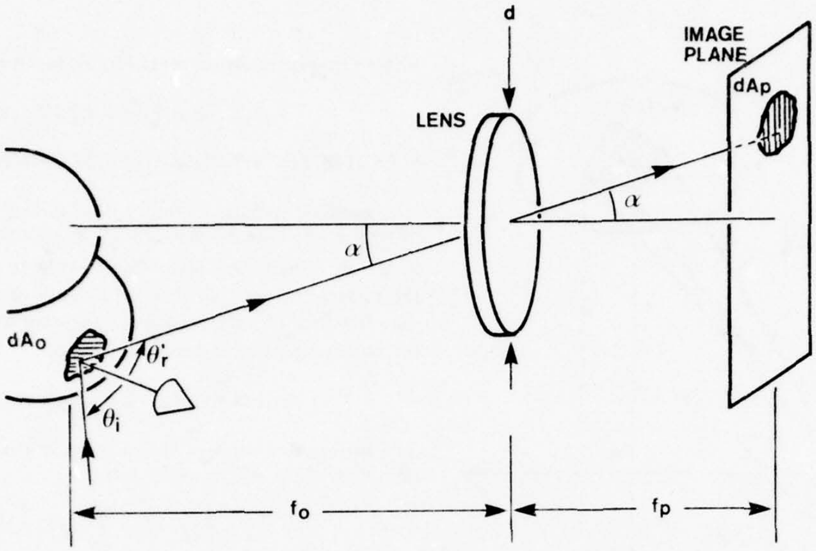


FIGURE 6: A simple image forming system. Light collected by the lens from the surface patch of area dA_o is projected into the image patch of area dA_p.

We will also need to know the area dA_p of the image of the patch, since image irradiance, E_p, is the flux per unit area:

$$E_p = d\Phi_L/dA_p$$

If θ'_r is the angle between the normal on the surface and the line to the entrance aperture nodal point, while α is the angle between this line and the optical axis, then, by equating solid angles,

$$(dA_o \cos \theta'_r)/f_o^2 = (dA_p \cos \alpha)/f_p^2$$

Consequently,

$$E_p = \cos \alpha \, (f_o/f_p)^2 \int L_r \, (\cos \theta_r/\cos \theta'_r) \, d\omega_r$$

Here the integral is over the solid angle occupied by the entrance aperture as seen from the patch on the surface. Note that θ_r in the integral will vary unless we assume that the lens is small relative to its distance from the object. In this case, θ_r is approximately the same as θ'_r, and can be cancelled. Furthermore, the reflected radiance, L_r, will tend to be constant and can be removed from the integral. The solid angle occupied by the lens as seen from the surface patch is approximately equal to the foreshortened area $(\pi/4)\, d^2 \cos \alpha$ divided by the distance $(f_o/\cos \alpha)$ squared. Finally then one obtains the well-known result,

$$E_p = (1/4)\,(d/f_p)^2 \cos^4\alpha \; \pi L_r$$

That is, image irradiance is proportional to scene radiance. The factor of proportionality is π divided by four times the square of the effective f-number (f_p/d), times the fourth power of the cosine of the off-axis angle, α. Thus the "sensitivity" of such an imaging system is not uniform but is constant for a particular point in the image. Vignetting introduces an additional variation with position in the image. Ideally an imaging device should be calibrated so that this variation in sensitivity as a function of α can be removed.

Other kinds of imaging systems, such as microscopes or mechanical scanners lead to somewhat different expressions. Generally, however, image irradiance is proportional to scene radiance in such systems. At this point we should remember that scene radiance depends on properties of the surface layer (BRDF) and the distribution of light-sources (source-radiance).

$$L_r = \int f_r \, L_i \, d\Omega_i$$

10. VIEWER-ORIENTED COORDINATE SYSTEM.

So far we have considered directions from the object to the image-forming system and to light sources in terms of a *local* coordinate system with one axis lined up with the surface normal. Such coordinate systems will vary in orientation from place to place and are thus inconvenient for the specification of global distributions such as that of source radiance. A coordinate system fixed in space will be more suitable, particulary if one of the axes is lined up with the optical axis (Fig. 7). In this *viewer-oriented* coordinate system we introduce polar angle, θ, measured from the z-axis and azimuth angle, ϕ, measured from the x-axis in the plane perpendicular to the z-axis. Directions to sources of light can be given using these two angles. If the sources are far away (in comparison to the size of the objects being imaged), then source radiance will be a fixed function of these angles independent of the point on the surface being considered.

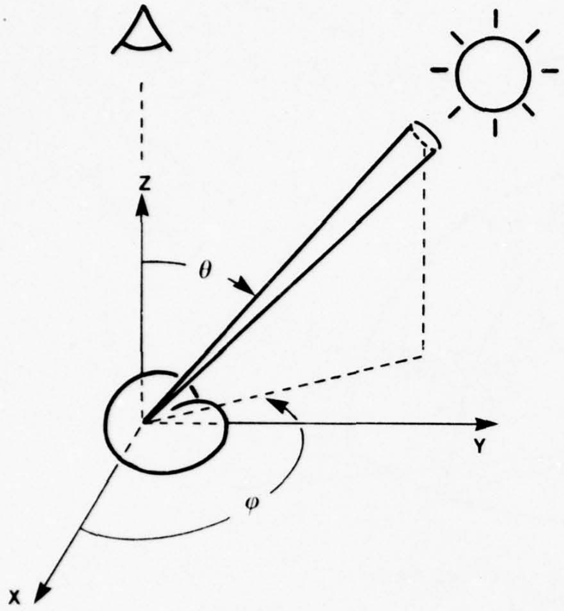


FIGURE 7: Viewer-oriented, global coordinate system useful for specification of the distribution of source radiance L_i.

11. THE SURFACE NORMAL

In the local coordinate system the surface normal is easily specified since it lies along one of the axes; or, equivalently, it is the direction corresponding to zero polar angle. In the viewer-oriented coordinate system the surface normal will correspond to some direction, say (θ_n, ϕ_n). The corresponding unit vector is,

$$\underline{n} = (\cos\phi_n \sin\theta_n,\ \sin\phi_n \sin\theta_n,\ \cos\theta_n)$$

The surface of the object may be specified by giving "elevation" z as a function of the coordinates x and y. We can give an expression for the surface normal in terms of the first partial derivatives of z with respect to x and y, if these exist. Let the first partial derivatives be called p and q. Then the vectors (1, 0, p) and (0, 1, q) are tangent to the surface, as can be seen by considering infinitesimal steps in the x and y direction. The surface normal is perpendicular to all vectors in the tangent plane and so is parallel to the cross-product of these two:

$$(1, 0, p) \times (0, 1, q) = (-p, -q, 1)$$

Thus the unit normal can be written

$$\underline{n} = (-p, -q, 1) \Big/ \sqrt{1+p^2+q^2}$$

The following results are obtained by equating terms in the two expressions for the surface normal:

$$\sin\theta_n = \sqrt{p^2+q^2} \Big/ \sqrt{1+p^2+q^2}$$
$$\cos\theta_n = 1/\sqrt{1+p^2+q^2}$$
$$\sin\phi_n = -q \Big/ \sqrt{p^2+q^2}$$
$$\cos\phi_n = -p \Big/ \sqrt{p^2+q^2}$$

Conversely,

$$p = -\cos\phi_n \tan\theta_n$$
$$q = -\sin\phi_n \tan\theta_n$$

12. RELATIONSHIP BETWEEN LOCAL AND VIEWER-ORIENTED COORDINATE SYSTEMS

In order to calculate the scene radiance we will integrate the product of the BRDF and the source radiance over all incident directions. Since the BRDF is specified in terms of the *local* coordinate system, while the distribution of source radiance is likely to be given in the *viewer-oriented* coordinate system, it will be necessary to convert between the two. Given the direction of the surface normal, (θ_n, ϕ_n), and the direction to a portion of the source (θ_s, ϕ_s), both specified in the viewer-oriented system (Fig. 8), we have to find the incident direction (θ_i, ϕ_i) and the exitant direction (θ_r, ϕ_r) both specified in the local system. Alternatively, given the surface normal and the incident direction we may have to find the direction to the source and the exitant direction. Note that $\theta_r = \theta_n$, since the exitant ray lies along the z-axis in the direction towards the viewer. Further, since we have excluded anisotropic surfaces, we are only interested in the *difference* between ϕ_r and ϕ_i. From the relevant spherical triangle (Fig. 9) we obtain

Cosine formula:

$$\cos\theta_i = \cos\theta_s \cos\theta_r + \sin\theta_s \sin\theta_r \cos(\phi_s - \phi_n)$$

Sine formula:

$$\sin\theta_i \sin(\phi_r - \phi_i) = \sin\theta_s \sin(\phi_s - \phi_n)$$

Analogue formula:

$$\sin\theta_i \cos(\phi_r - \phi_i) = \cos\theta_s \sin\theta_r - \sin\theta_s \cos\theta_r \cos(\phi_s - \phi_n)$$

The Jacobian of the transformation from (θ_s, ϕ_s) to (θ_i, ϕ_i) equals,

$$(\partial\theta_i/\partial\theta_s)(\partial\phi_i/\partial\phi_s) - (\partial\theta_i/\partial\phi_s)(\partial\phi_i/\partial\theta_s) = (\sin\theta_s / \sin\theta_i).$$

The above formulae allow us to find the incident direction from the source direction. Quite symmetrically, we can also obtain the source direction from the incident direction:

Cosine formula:

$$\cos\theta_s = \cos\theta_i \cos\theta_r + \sin\theta_i \sin\theta_r \cos(\phi_r - \phi_i)$$

Sine formula:

$$\sin\theta_s \sin(\phi_s - \phi_n) = \sin\theta_i \sin(\phi_r - \phi_i)$$

Analogue formula:

$$\sin\theta_s \cos(\phi_s - \phi_n) = \cos\theta_i \sin\theta_r - \sin\theta_i \cos\theta_r \cos(\phi_r - \phi_i)$$

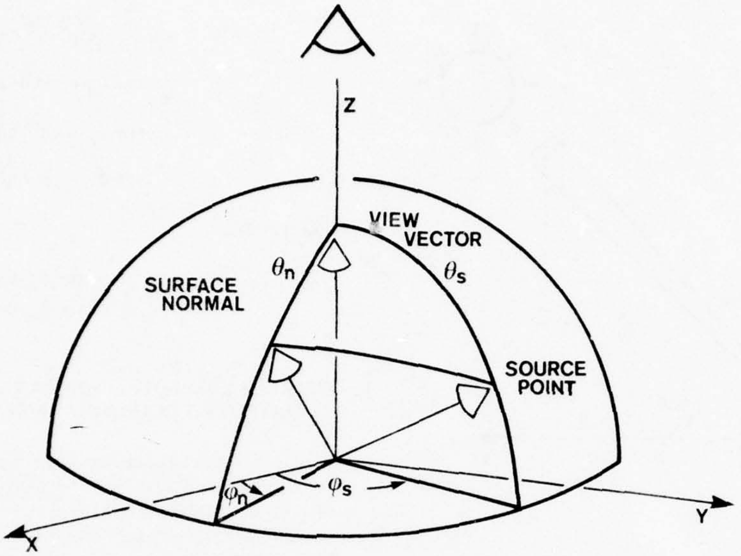


FIGURE 8: Surface normal and direction to portion of the source shown in viewer-oriented coordinate system.

The Jacobian of the transformation from (θ_i, ϕ_i) to (θ_s, ϕ_s) equals

$$(\partial\theta_s/\partial\theta_i)\,(\partial\phi_s/\partial\phi_i) - (\partial\phi_s/\partial\theta_i)\,(\partial\theta_s/\partial\phi_i) = (\sin\theta_i \,/\, \sin\theta_s).$$

13. SCENE RADIANCE

It follows from the definition of the BRDF that reflected radiance can be written as the integral

$$L_r = \int f_r\, L_i\, d\Omega_i = \int f_r\, L_i \cos\theta_i\, d\omega_i$$

Using polar and azimuthal angles this becomes

$$L_r(\theta_n, \phi_n) = \int_{-\pi}^{\pi} \int_0^{\pi/2} f_r(\theta_i, \phi_i; \theta_r, \phi_r)\, L_i(\theta_s, \phi_s) \cos\theta_i \sin\theta_i\, d\theta_i\, d\phi_i$$

Here we integrate over all possible incident directions (θ_i, ϕ_i) and calculate source direction (θ_s, ϕ_s) from the given surface normal (θ_n, ϕ_n) and the incident direction. The inner integral has the limits 0 to $\pi/2$ for θ_i, corresponding to directions within the hemisphere visible from the surface. The integration can be extended to the full sphere of directions if the integrand is forced to be zero when θ_i lies between $\pi/2$ and π. This can be accomplished by replacing $\cos\theta_i$ by $\max[0, \cos\theta_i]$. Hence

$$L_r = \int_{-\pi}^{\pi} \int_0^{\pi'} f_r\, L_i \max[0, \cos\theta_i] \sin\theta_i\, d\theta_i\, d\phi_i$$

Since the integral now is over the full sphere of directions, it can be rewritten using any other set of polar and azimuth angles. Using the *viewer-oriented* coordinate system for example we obtain,

$$L_r = \int_{-\pi}^{\pi} \int_0^{\pi} f_r\, L_i \max[0, \cos\theta_i] \sin\theta_s\, d\theta_s\, d\phi_s$$

That is,

$$L_r(\theta_n, \phi_n) = \int_{-\pi}^{\pi} \int_0^{\pi} f_r(\theta_i, \phi_i; \theta_r, \phi_r)\, L_i(\theta_s, \phi_s) \max[0, \cos\theta_i] \sin\theta_s\, d\theta_s\, d\phi_s$$

Here we integrate over all possible source directions (θ_s, ϕ_s) and calculate incident directions (θ_i, ϕ_i) from the given surface normal (θ_n, ϕ_n) and the source direction. We now have two convenient forms for the calculation of scene radiance. We proceed to calculate reflectance maps for a few simple combinations of BRDF and distributions of source radiance.

14. COLLIMATED SOURCE, LAMBERTIAN REFLECTANCE

For a lambertian reflector, $f_r = 1/\pi$. For a collimated source,

$$L_i = E_0\, \delta(\theta_s - \theta_0)\, \delta(\phi_s - \phi_0) \,/\, \sin\theta_0$$

where E_0 is the irradiance measured perpendicular to the beam of light arriving from source direction (θ_0, ϕ_0). Substituting into the second form of the expression for scene radiance above, we obtain

$$L_r = \int_{-\pi}^{\pi} \int_0^{\pi} (E_0/\pi)\, \delta(\theta_s - \theta_0)\, \delta(\phi_s - \phi_0) \cdot \max[0, \cos\theta_i]\, (\sin\theta_s \,/\, \sin\theta_0)\, d\theta_s\, d\phi_s$$

this becomes,

$$L_r = (E_0/\pi) \max[0, \cos \theta_i]$$

where

$$\cos \theta_i = \cos \theta_r \cos \theta_0 + \sin \theta_r \sin \theta_0 \cos(\phi_0 - \phi_n)$$

Note that

$$\cos(\phi_0 - \phi_n) = \cos \phi_0 \cos \phi_n - \sin \phi_0 \sin \phi_n$$

To obtain the reflectance map, scene radiance as a function of surface gradient, we can substitute expressions in p and q for these trigonometric expressions. The result is

$$R(p, q) = (E_0/\pi) \max \left(0, \; 1+p_0p+q_0q \Big/ \left(\sqrt{1+p^2+q^2}\sqrt{1+p_0^2+q_0^2}\,\right)\right)$$

where

$$p_0 = -\cos \phi_0 \tan \theta_0$$
$$q_0 = -\sin \phi_0 \tan \theta_0$$

The significance of p_0 and q_0 is that a surface element with gradient (p_0, q_0) has its surface normal parallel to the direction of the incident light rays.

15. UNIFORM SOURCE, LAMBERTIAN REFLECTANCE

A uniform source has constant incident radiance. Let $L_s = L_0$. Again, for a lambertian reflector, $f_r = 1/\pi$. Substituting into the first form of the expression for scene radiance, we obtain

$$L_r = \int_{-\pi}^{\pi} \int_{0}^{\pi/2} (L_0/\pi) \cos \theta_i \sin \theta_i \, d\theta_i \, d\phi_i$$

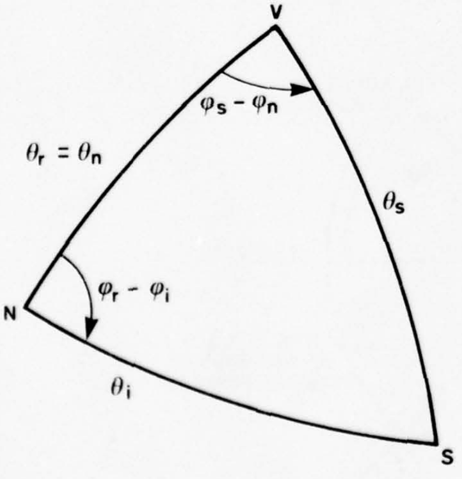


FIGURE 9. Spherical triangle extracted from previous figure and used in derivation of transformation equations between the surface-normal, local coordinate system and the viewer-oriented global coordinate system.

This becomes

$$L_r = L_0 \int_{0}^{\pi/2} \sin 2\theta_i \, d\theta_i = L_0$$

Not surprisingly, the reflected radiance is independent of the surface orientation in this case.

16. COLLIMATED SOURCE, SPECULAR REFLECTANCE

For specular surfaces,

$$f_r = \delta(\theta_i - \theta_r)\, \delta(\phi_i - \phi_r + \pi) / (\sin \theta_i \cos \theta_i)$$

Using the source radiance from section 14 above, and the first form of the expression for scene radiance we obtain,

$$L_r = \int \int (E_0 / \sin \theta_0)\, \delta(\theta_i - \theta_r)\, \delta(\phi_i - \phi_r + \pi)\, \delta(\theta_s - \theta_0)\, \delta(\phi_s - \phi_0)\, d\theta_i \, d\phi_i$$

That is,

$$L_r = E_0\, \delta(\theta_s' - \theta_0)\, \delta(\phi_s' - \phi_0) / \sin \theta_0$$

where θ_s' and ϕ_s' are the values of θ_s and ϕ_s corresponding to $\theta_i = \theta_r$ and $\phi_i = \phi_r + \pi$. Using the equations for the coordinate transformations one finds that $\theta_s' = 2\,\theta_r$ and $\phi_s' = \phi_n$. Thus,

$$L_r = E_0\, \delta(2\,\theta_r - \theta_0)\, \delta(\phi_n - \phi_0) / \sin \theta_0$$

Or finally,

$$L_r(\theta_n, \phi_n) = (E_0/2)\, \delta(\theta_n - \theta_0/2)\, \delta(\phi_n - \phi_0) / \sin \theta_0$$

To express this as a function of p and q we have to remember that

$$\delta[f(x, y) - f(x_0, y_0)]\, \delta[g(x, y) - g(x_0, y_0)]$$
$$= \delta(x - x_0)\, \delta(y - y_0) / J(x_0, y_0)$$

where $J(x,y)$ is the Jacobian of the transformation from (x,y) to (f,g):

$$J(x,y) = (\partial f/\partial x)(\partial g/\partial y) - (\partial f/\partial y)(\partial g/\partial x)$$

The Jacobian of the transformation from (p, q) to (θ_n, ϕ_n) is

$$J(p, q) = 1/[\sqrt{p^2 + q^2}\;(1 + p^2 + q^2)]$$

Let

$$p_1 = -\cos \phi_0 \tan \theta_0/2 \quad \text{and} \quad q_1 = -\sin \phi_0 \tan \theta_0/2.$$

Then, noting that $\sin \theta_0 = 2 \sin \theta_0/2 \cos \theta_0/2$, one can write

$$\sin \theta_0 = 2 \sqrt{p_1^2 + q_1^2} \Big/ (1 + p_1^2 + q_1^2)$$

and therefore,

$$R(p, q) = (1/4)\, \delta(p - p_1)\, \delta(q - q_1)\, (1 + p_1^2 + q_1^2)^2$$

The significance of p_1 and q_1 is that a surface element with gradient (p_1, q_1) is oriented to specularly reflect the collimated source towards the viewer. This gradient can be related to the gradient (p_0, q_0) introduced earlier.

$$p_1 = p_0 \left(\sqrt{1+p_0^2+q_0^2} - 1\right) / (p_0^2+q_0^2)$$
$$q_1 = q_0 \left(\sqrt{1+p_0^2+q_0^2} - 1\right) / (p_0^2+q_0^2)$$

The point (p_1, q_1) is approximately half as far from the origin, (0, 0), as the point (p_0, q_0), when the latter is not too far from the origin.

17. UNIFORM SOURCE, SPECULAR REFLECTANCE

It is easy to see that for a specular surface under a uniform source, the scene radiance will be constant and equal to the source radiance.

$$L_r = L_0$$

This is the same result as the one we obtained for the uniform source and lambertian reflectance. Thus a diffuse surface appears just as bright as a specular surface if both are viewed with uniform illumination. In fact, all surfaces reflecting the same fraction, ρ say, of the total incident light will appear equally bright under this illumination condition.

18. HEMISPHERICAL UNIFORM SOURCE, LAMBERTIAN REFLECTANCE

A hemisherical uniform source is described by

$$L_i(\theta_s, \phi_s) = L_0 \quad \text{for } \theta_s < 90^\circ$$
$$L_i(\theta_s, \phi_s) = 0 \quad \text{for } \theta_s > 90^\circ$$

To evaluate the double integral for scene radiance, it is helpful to know the value θ_i' of θ_i which corresponds to the horizon $\theta_s = \pi/2$. From the coordinate transformation equations one can easily show that

$$\cot \theta_i' = - \tan \theta_r \cos (\phi_r - \phi_i)$$

For $\phi_r - \pi/2 < \phi_i < \phi_r + \pi/2$, the horizon cutoff will occur for $\theta_i' > \pi/2$ and can be ignored. For the other half of the range of ϕ_i, this cutoff occurs for $\theta_i' < \pi/2$ and must be considered. Now,

$$\int_0^{\pi/2} \cos \theta_i \sin \theta_i \, d\theta_i = 1/2$$

while

$$\int_0^{\theta_r'} \cos \theta_i \sin \theta_i \, d\theta_i = (1 - \cos 2\,\theta_i') / 4 = \sin^2\theta_i' / 2$$

If $\cot \theta_i' = - \tan \theta_r \cos(\phi_r - \phi_i)$, then

$$\sin^2\theta_i' = 1 / [1 + \tan^2\theta_r \cos^2(\phi_r - \phi_i)]$$

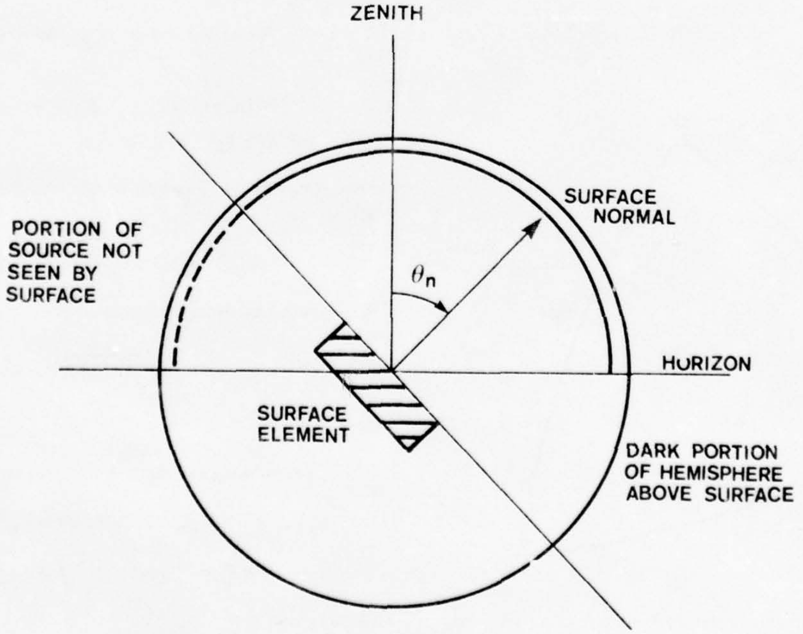

ZENITH
SURFACE NORMAL
PORTION OF SOURCE NOT SEEN BY SURFACE
θ_n
HORIZON
SURFACE ELEMENT
DARK PORTION OF HEMISPHERE ABOVE SURFACE

FIGURE 10: Cross-section through uniform hemispherical source and surface element, illustrating horizon cutoff and portion of extended source not visible from surface.

Now,

$$L_r = \int_{-\pi}^{\pi} \int_{c}^{\pi/2} f_r\, L_i \cos \theta_i \sin \theta_i \, d\theta_i \, d\phi_i$$

Substituting for f_r and L_i and splitting the range of integration, we get

$$L_r = (L_0/2\pi) \left[\int_{-\pi}^{\phi_r - \pi/2} \sin^2\theta_i' \, d\phi_i + \int_{\phi_r - \pi/2}^{\phi_r + \pi/2} d\phi_i + \int_{\phi_r + \pi/2}^{\pi} \sin^2\theta_i' \, d\phi_i \right]$$

The integral in the middle is just equal to π, while the outer two integrals add up to

$$\int_{-\pi/2}^{\pi/2} 1 / [1 + \tan^2\theta_r \cos^2\phi] \, d\phi$$

which equals

$$[\cos \theta_r \tan^{-1} (\cos \theta_r \tan \phi)] \quad = \pi \cos \theta_r$$

Adding up all the terms we finally get,

$$L_r(\theta_n, \phi_n) = L_0 (1 + \cos \theta_n) / 2 = L_0 \cos^2 \theta_n/2$$

This is the result found by Brooks [44]. From it one can immediately determine the reflectance map

$$R(p, q) = (L_0/2) (1 + 1/\sqrt{1+p^2+q^2})$$

SUMMARY AND CONCLUSIONS

We have shown that image irradiance is proportional to scene radiance and that scene radiance depends on surface orientation. The reflectance map gives scene radiance as a function of the gradient. It can be calculated from the bidirectional reflectance-distribution function (BRDF) and the distribution of source radiance. Several special cases were worked out in detail. Each could have been developed more easily by a direct method, but was obtained from the general expression for scene radiance to illustrate the technique. The general expression allows one to find the reflectance map even if the source radiance distribution or the BRDF is only given numerically.

ACKNOWLEDGEMENTS

The authors would like to thank H.-H. Nagel, B. Neumann, B. Radig and L. Dreschler of the University of Hamburg, who helped debug some of the ideas. Discussions with M. Brady and M. Brooks of the University of Essex were very helpful. We are also grateful for text preparation by Blenda Horn and drawings by Karen Prendergast.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643.

REFERENCES

[1] Diggelen, J. van (1951) "A Photometric Investigation of the Slopes and Heights of the Ranges of Hills in the Maria of the Moon," *Bulletin of the Astronomical Institute of the Netherlands*, July 1951.

[2] Rindfleisch, T. (1966) "Photometric Method for Lunar Topography," *Photogrammetric Engineering*, March 1966.

[3] Horn, B. K. P. (1975) "Determining Shape from Shading," Chapter 4, in Winston, P. H. (Ed) *The Psychology of Computer Vision*, McGraw-Hill, New York.

[4] Horn, B. K. P. (1977) "Understanding Image Intensities," *Artificial Intelligence*, Vol. 8, No. 11, pp 201-231.

[5] Woodham, R. J. (1977) "A Cooperative Algorithm for Determining Surface Orientation from a Single View," *Proc. 5th Int. Joint Conf. on Artif Intell.*, M. I. T., Cambridge, Massachusetts, August 1977, pp 635-641.

[6] Woodham, R. J. (1978) "Reflectance Map Techniques for Analyzing Surface Defects in Metal Castings," TR-457, A. I. Laboratory, M. I. T., Cambridge, Massachusetts.

[7] Woodham, R. J. (1978) "Photometric Stereo: A Reflectance Map Technique for Determining Surface Orientation from Image Intensity", *Proceedings of SPIE's 22nd Annual Technical Symposium*, Vol. 155, August 1978.

[8] Horn, B. K. P. and Woodham, R. J. (1978) "Determining Shape and Reflectance Using Multiple Images," A. I. Memo 490, M. I. T., Cambridge, Massachusetts. August 1978.

[9] Gilpin, F. H. (1910) "Effect of the variation of the incident angle on the coefficients of diffuse reflection," *Trans. Illum. Eng. Soc.*, Vol. 5, pp 854-873.

[10] Knowles Middleton, W. E. and Mungall, A. G. (1952) "The luminous directional reflectance of snow,"*Journal of the Optical Society of America*, Vol. 42, No. 8, August 1952, pp 572-579.

[11] Fedoretz, V. A. (1952) "Photographic Photometry of the Lunar Surface," *Publ. Kharkov Obs.*, Vol. 2, pp 49-172.

[12] Van Diggelen, J. (1959) "Photometric Properties of Lunar Crater Floors," *Rech. Obs. Utrecht*, Vol. 14, pp 1-114.

[13] Minnaert, M. (1961) "Photometry of the Moon," in *Planets and Satellites*, G. Kuiper and B. Middlehurst (eds), Vol. 3, Univ. of Chicago Press, Chicago, pp 213-248.

[14] Fesenkov, V. (1962) "Photometry of the Moon," in *Physics and Astronomy of the Moon*, Z. Kopal (ed), Academic Press, New York, pp 99-130.

[15] Hapke, B. and Van Horn, H. (1963) "Photometric Studies of Complex Surfaces, with Applications to the Moon," *Journal of Geophysical Research*, Vol. 68, No. 15, pp 4545-4570.

[16] De Vaucouleurs, G. (1964) "Geometric and Photometric Parameters of the Terrestrial Planets," *ICARUS*, Vol. 3, pp 187-235.

[17] Van Diggelen, J. (1965) "The radiance of lunar objects near opposition," *Planetary Space Science*, Vol. 13, pp 271-279.

[18] Oetking, P. (1966) "Photometric Studies of Diffusely Reflecting Surfaces with Applications to the Brightness of the Moon," *Journal of Geophysical Research*, Vol. 71, No. 10, May 1966, pp 2505-2515.

[19] Rennilson, J. J., Holt, H. E. and Morris, E. C. (1968) "*In Situ* Measurements of the Photometric Properties of an Area on the Lunar Surface," *Journal of the Optical Society of America*, Vol. 58, No. 6, June 1968, pp 747-755.

[20] Patterson, E. M., Sheldon, C. E. and Stockton, B. H. (1977) "Kubelka-Munk optical properties of a barium surfate white reflectance standard," *Applied Optics*, Vol. 16, No. 3, March 1977, pp 729-732.

[21] Tucker, C. J. (1977) "Asymptotic nature of grass canopy spectral reflectance," *Applied Optics*, Vol. 16, No. 5, May 1977, pp 1151-1156.

[22] Minnaert, M. (1941) "The Reciprocity Principle in Lunar Photometry," *Astrophysical Journal*, Vol. 93, pp 403-410.

[23] Van de Hulst, H. C. (1957) *Light Scattering by Small Particles*," John Wiley and Sons, New York.

[24] Hapke, B. W. (1963) "A Theoretical Photometric Function for the Lunar Surface," *Journal of Geophysical Research*, Vol. 68, No. 15, August 1963, pp 4571-4586.

[25] Melamed, N. T. (1963) "Optical Properties of Powders. Part I. Optical Absorption Coefficients and the Absolute Value of Diffuse Reflectance. Part II. Properties of Luminescent Powders," *Journal of Applied Optics*, Vol. 34, No. 3, March 1963, pp 560-570.

[26] Hapke, B. (1966) "An Improved Theoretical Lunar Photometric Function," *The Astronomical Journal*, Vol. 71, No. 5, June 1966, pp 333-339.

[27] Torrance, K. E., Sparrow, E. M. and Birkebak, R. C. (1966) "Polarization, directional distribution, and off-specular peak phenomena in light reflected from roughened surfaces," *Journal of the Optical Society of America*, Vol. 56, No. 7, July 1966, pp 916-925.

[28] Torrance, K. E. and Sparrow, E. M. (1967) "Theory for off-specular reflection from roughened surfaces," *Journal of the Optical Society of America*, Vol. 57, No. 9, September 1967, pp 1105-1114.

[29] Trowbridge, T. S. and Reitz, K. P. (1975) "Average irregularity representation of a rough surface for ray reflection", *Journal of the Optical Society of America*, Vol. 65, No. 5, May 1975, pp 531-536.

[30] Simmons, E. L. (1975) "Diffuse reflectance spectroscopy: a comparison of the theories," *Applied Optics*, Vol. 14, No. 6, June 1975, pp 1380-1336.

[31] Simmons, E. L. (1975) "A refinement of the simplified particle model theory of diffuse reflectance spectroscopy," *Optica Acta*, Vol. 22, No. 1, pp 71-77.

[32] Simmons, E. L. (1975) "Modification of the particle-model theory of diffuse reflectance properties of powdered samples," *Journal of Applied Physics*, Vol. 46, No. 1, January 1975, pp 344-348.

[33] Simmons, E. L. (1976) "Particle model theory of diffuse reflectance: effect of nonuniform particle size," *Applied Optics*, Vol. 15, No. 3, March 1976, pp 603-604.

[34] Tucker, C. J. and Garratt, M. W. (1977) "Leaf optical system modelled as a stochastic process," *Applied Optics*, Vol. 16, No. 3, March 1977, pp 635-642.

[35] Plass, G. N., Kattawar, G. W. and Guinn Jr., J. A. (1977) "Isophotes of sunlight glitter on a wind-ruffled sea," *Applied Optics*, Vol. 16, No. 3, March 1977, pp 643-653.

[36] Gouraud, H. (1971) "Computer display of curved surfaces," Technical Report 113, UTEC-CSC-71, Computer Science, University of UTAH, Salt Lake City, Utah.

[37] Bui Tuong-Phong (1973) "Illumination for computer-generated images," Technical Report 129, UTEC-CSC-73, Computer Science, University of Utah, Salt Lake City, Utah.

[38] Blinn, J. F. (1977) "Models of light reflection for computer synthesized pictures," *SIGGRAPH '77, Proc. ACM, Computer Graphics*, Vol. 11, No. 2, July 1977, pp 192-198.

[39] Wendlandt, W. W. and Hecht, H. G. (1966) *Reflectance Spectroscopy*, Interscience, New York.

[39] Wendlandt, W. W. (ed) (1968) *Modern Aspects of Reflectance Spectroscopy*, Plenum, New York.

[41] Kortuem, G. (1969) *Reflectance Spectroscopy*, trans. by J. E. Lohr, Springer-Verlag, Berlin.

[42] Spencer, D. E. and Gaston, E. G. (1975) "Current definitions of reflectance," *Journal of the Optical Society of America*, Vol. 65, No. 10, October 1975, pp 1129-1132.

[43] Nicodemus, F. E., Richmond, J. C. and Hsia, J. J., Ginsberg, I. W. and Limperis, T. (1977) "Geometrical Considerations and Nomenclature for Reflectance," NBS Monograph 160, National Bureau of Standards, U. S. Department of Commerce, Washington, D. C., October 1977.

[44] Brooks, M. J. (1978) "Investigating the Effects of Planar Light Sources," CSM 22, Dept. of Computer Science, Essex University.

ROBUST PICTURE PROCESSING OPERATORS AND THEIR IMPLEMENTATION AS CIRCUITS

Michael Ian Shamos

Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213

ABSTRACT

The increasing use of satellites in image acquisition has made real-time data compression and summary essential. To reduce bandwidth and alleviate the load on land-based computers it is desirable to perform as much picture processing as possible via LSI circuitry aboard the satellite. Such circuits must be able to deal with a wide variety of images and must exhibit a high degree of reliability. In this paper we use some results from the theory of selection networks to produce a family of robust image smoothing operators suitable for LSI implementation. The circuits are (1) decomposable into small functional units, (2) easily testable, and (3) statistically insensitive to spikes or noise in the data.

1. Introduction

(The entire problem treated in this paper was suggested by Prof. Raj Reddy.) One technical difficulty in current image processing is that resolutions are so high that we literally are unable to see the forest for the trees. It is important to be able to "defocus" minute details to become aware of the larger object of which they are a part. A separate problem is to compress or summarize the image to reduce the telecommunications burden. The encoded picture will then be reconstructed on the ground and it is crucial to extract statistics that suffice to perform this task. Our purpose here is to suggest a new method by which this defocusing and compression may be accomplished.

2. Median Smoothing

In what follows we will assume that an "image" consists of a rectangular array of grey-scale intensities. Our operators will operate on n-by-n square submatrices of the image, where n is odd and small (typically n = 3 or 5). The function of the operator is to compute a descriptive statistic of the n^2 pixels on which it acts. Let F(i,j) denote the value of this statistic over the n-by-n submatrix centered at position (i,j) in the original image array I. One method of smoothing the image that is useful for detecting gross objects is to replace each element I(i,j) by F(i,j). (If F were the averaging operator, for example, then this would correspond to taking moving averages.) One may also effect data compression by a factor of n^2 by replacing the entire submatrix centered at I(i,j) by the single value F(i,j) whenever i and j are congruent to (n+1)/2 modulo n. This procedure can be applied recursively to produce a sequence of progressively defocused (blurred) images. For example, if I is a 625-by-625 matrix, then applying this operation once will yield a 25-by-25 matrix and applying it a second time will give a 5-by-5 result.

Which choices for the smoothing operator F are suitable for picture processing applications? It should possess at least the following properties:

a) F should be robust, that is, it should be relatively insensitive to outlying values, or spikes. (These may correspond to bright spots, reflections, or damaged areas on the retina.)

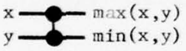
b) F(i,j) should equal at least one of the actual values in the submatrix on which it operates. This condition is imposed because if the submatrix contains parts of two or more objects, we would like F to serve as a descriptor for the object that occupies "most" of the submatrix. For example, in the subimage at the left we have pieces of two objects, with intensities 1 and 3. We wish F to reflect the fact that the subimage is composed primarily of part of object 3.

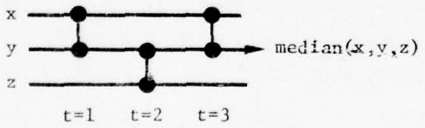
1	1	1	3	3
1	1	3	3	3
1	3	3	3	3
3	3	3	3	3
3	3	3	3	3

The averaging operator (mean) possesses neither of these two properties, but the median possesses both. In the next section we will attempt to design a circuit for computing the median but will compromise instead on an approximation to the median that is more suitable on several grounds for LSI implementation.

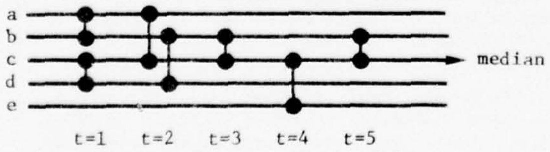
3. Circuits for the median and approximations

The chief difficulty in computing the median of $M = n^2$ quantities is that it is not an algebraic function of the M inputs and cannot be calculated using arithmetic operations alone -- comparisons are required. It is possible, in fact to compute the median using only comparisons. To show how to implement such algorithms as circuits, we will make use of comparator modules and selection networks, as described in [1]. A comparator module is a device with two input lines, x and y, and two output lines, as shown below. The input signals are compared and the larger is routed to the upper output line, while the smaller appears at the lower output line. In our diagrams of comparator networks, signals will be assumed to enter from the left and exit at the right. The following network finds the median of three inputs using three comparators and a time delay of three:





(The above circuit actually sorts its inputs.) It may not be readily apparent, but the next network finds the median of five inputs:



The median-of-5 network exploits parallelism during the first two time steps to achieve an overall delay of five with seven comparators. It is shown in [1] that the number of comparators cannot be reduced. We have shown by exhaustion that a delay of five is optimal for comparator networks with fanout one.

There are analogous networks for larger sets of inputs but they become progressively more complex and difficult to design. We do not know how to construct networks that are optimal either with respect to time delay or number of comparators for any but the smallest values of M. Furthermore, the structure of near-optimal circuits is highly irregular and not readily decomposable into simple functional units. A problem that looms larger, however, is that of testability. Once a network is constructed, either theoretically or in practice, how can we verify that it works? If each of the M inputs can assume any of C possible distinct values, it would seem that C^M separate tests are required. For a circuit consisting solely of comparators, through, it suffices to verify its correctness when each input is restricted to be either zero or one. This result is known as the 0-1 Principle [1] and it reduces the number of tests required to just 2^M. While this is a significant improvement, even if we are able to design a median network for 5-by-5 submatrices, verifying all 2^{25} possible binary inputs would be out of the question. To circumvent this difficulty, we will explore an alternative to the exact median which has excellent statistical properties, is decomposable, and is easily tested.

To obtain an approximation to the median we will generalize an idea due to Tukey [2]. For M = 9, let us compute p = median(a,b,c), q = median(d,e,f), and r = median(g,h,i). Now let s = median(p,q,r), that is, the median of the medians. If we implement this computation via a comparator network, it is easy to see that p,q, and r can all be found in parallel in three time steps using nine comparators by replicating the median-of-3 circuit at the left three times. The quantity s can then be found with three more comparators and three additional time steps by using a fourth copy of this circuit in an elegant cascade arrangement. The total number of comparators is 12 and the time delay is six. (The number of comparators can be reduced to nine by re-using one of the first three median circuits.) For M = 25 a similar partitioning into medians of five gives a circuit with 35 comparators and a delay of 10 that can be tested by trying only $5 \cdot 2^5 = 160$ different inputs as opposed to 2^{25}.

The cascade median circuit can be generalized directly for arbitrary odd values of n, the number of tests required being $n2^n$ for n^2 inputs. However, it must be emphasized that these circuits do not compute the median, but only some approximation to the median. We will now investigate how good this approximation is. Let A_n denote the cascade median as found above. If n = 3, then we are trying to find the median of nine elements, that is, the element that has rank five. It is shown in [2] that if all 9! permutations of the inputs are equally likely then A_3 is the exact median (rank 5) with probability 4/7, or approximately 0.571. A_3 will have rank four or rank six with equal probabilities 3/14. Determining the distribution of A_n, even under the assumption of equal probability for each permutation (an assumption that can be relaxed somewhat), is a difficult combinatorial problem. For n = 5 (25 elements) it was easier to obtain the distribution by simulating 100,000 cases than by attempting an exact calculation. The results of the simulation are given below. The exact median has rank 13 out of 25.

$$\begin{aligned}
P(\text{rank} = 9) &= P(\text{rank} = 17) \approx 0.0052\\
P(\text{rank} = 10) &= P(\text{rank} = 16) \approx 0.0313\\
P(\text{rank} = 11) &= P(\text{rank} = 15) \approx 0.1023\\
P(\text{rank} = 12) &= P(\text{rank} = 14) \approx 0.2162\\
P(\text{rank} = 13) &\approx 0.2900
\end{aligned}$$

Distribution of A_5 (obtained by simulation)

Thus A_5 is the exact median with probability 0.29 and has rank that is within one of the correct median with probability > 0.72. We see that A_5 is strongly peaked about the true median. It is clear also from the symmetry of the algorithm that the expected rank of A_n is $(n^2+1)/2$, that is, the true median. We now show that A_n is guaranteed to filter out the upper and lower quartiles of the data completely.

Theorem. $\text{rank}(A_n) \geq (n^2 + 2n + 1)/4$ and
$\text{rank}(A_n) \leq (3n^2 - 2n + 3)/4$.

Proof: To obtain the first inequality we need only observe that A_n surely exceeds (n + 1)/2 of the values in (n - 1)/2 of the n-sets and (n - 1)/2 of the values in its own n-set. The proof of the second inequality is analogous. □

In summary, A_n has the following desirable properties:

a) It is unbiased for the median.

b) It is strongly concentrated about the median.

c) It is outlier-resistant because the upper quarter and lower quarter of the data are eliminated completely.

We contend that A_n is an easily-computable and admirable substitute for the median in picture processing applications.

4. Extensions and Unsolved Problems

The mean and variance are sufficient statistics for normally-distributed data. Their robust analogs are the median and interquartile range, respectively. (The interquartile range is the difference between the first and third quartile values.) It would be useful to generalize the cascade circuits so that they produce an estimate of the interquartile range. One would also like to obtain the exact distribution of A_n and the interquartile range estimate.

The comparator modules discussed in this paper are not ideal for LSI implementation and the median computation can be performed using more suitable primitives. The methods presented here, however, at least illustrate the theoretical tools that one might use to design and test an actual implementation. Other probabilistic approaches are suggested in [3].

5. Acknowledgements

The author wishes to thank Professors Jon Bentley and Bruce Weide for engaging discussions. This research was supported in part by the Office of Naval Research under Contract N00014-76-C-0373.

REFERENCES

[1]. Knuth, D. E., The Art of Computer Programming, Volume III: Sorting and Searching, Addison-Wesley (1973). Section 5.3.4.

[2]. Tukey, J. W., The ninther, a technique for low effort robust(resistant) location in large samples, in David, H. A., ed., Contributions to Survey Sampling and Applied Statistics, Academic Press (1978), pp. 251-257.

[3]. Weide, B. W., Statistical Methods in Algorithm Design and Analysis, Ph.D. Thesis, Department of Computer Science, Carnegie-Mellon University (1978), unpublished.

INVERSE SYNTHETIC APERTURE RADAR IMAGING

by

Harry C. Andrews
Chung-Ching Chen

Image Processing Institute
University of Southern California, Los Angeles, California 90007

ABSTRACT

Imaging from ground-based (stationary) radars of moving targets is often possible by utilizing a "synthetic aperture" developed from the target motion itself. An aircraft is imaged from both a straight flight and a turn with recognizable results. Analysis shows that two phase components exist in the radar return, one being gross velocity induced, the other being interscatterer interference within the target itself. The former phase must be removed prior to imaging and techniques are developed for this task. Coherence processing intervals, range collapsing, and range re-alignment are all examined herein.

INTRODUCTION

In order to reconstruct a radar image of some target from its signal returns, two conditions have to be satisfied. First, the returned data has to have some kind of two-dimensional format. Second, the radar imaging geometry must be such that the return from each pulse or signature contains different (could be only "slight") information about the target.

In the usual case of a pulsing radar, the return from a single pulse contains timing or range information, while the direction, called azimuth, along which the many pulse returns are aligned side by side, contains cross range information, and thus the first requirement for imaging is readily met. The second requirement demands that each pulse return be different. To accomplish this it is necessary to create a relative motion between the target and radar in such a way that the aspect angles of the target as observed from the radar are different for different pulses so that the cross range or azimuth information can be inferred. In this report, we look into a ground-based radar system in which a target aircraft is imaged by its own motion induced doppler.

Figure 1 shows the flight path of a target aircraft which has an overall length of approximately 80 feet and wing span of about 70 feet. Two portions of the flight path along which the data were obtained for imaging will be called interval 1 and interval 2, as shown in Fig. 1. The first interval is when the airplane was flying straight, at angles approximately 30 to 15 off broadside, whereby the second interval occurs when the airplane was making a standard left turn.

PREPROCESSING

For most practical purposes, the radar imaging system which determines the relation between the data returns and the reflectivities of the target can be considered linear [1], [2] and the system classification method developed elsewhere can be used to decide the ways to reconstruct the reflectivities directly from the raw data. This situation is depicted in Fig. 2. In other words, the data return $g(x,y)$ is a linear transformation of target reflectivity function $f(\xi,\eta)$ through the radar signal radiation and the echo reception. For ease of presentation we will assume that both g and f in Fig. 2 are discrete so that the system can be represented by a matrix [H] and g and f are vectors [3]. Depending on the waveforms of transmitted signals, (e.g. short pulse, linear FM pulse, or step-frequency waveforms) and the imaging geometries (e.g., shape and size of target, direction of relative motion, resolution required, etc.), the radar imaging systems represent a wide spectrum of the classes. Once the relation [H] between the reflectivity and data is (precisely) decided by the flight or radar data, a straightforward reconstruction of $\underline{f}$ and $\underline{g}$ can be achieved by applying the pseudoinverse of [H] to $\underline{g}$ yielding a minimum square error reconstruction.

The above reconstruction scheme, although straightforward in theory, usually involves a great deal of computation because of the complexity of [H]. In the worse case, one would expect to resort to a full singular value decomposition (SVD) to find $[H]^{-1}$. Certainly a decomposition of [H] such that the structure of the imaging geometry can be better utilized would warrant the efforts in many cases.

A perceivable way to accomplish this is to do some preprocessing upon the raw data such that the resultant data have a much simplified relation to the reflectivity than the raw data itself. Diagrammatically, [H] can be replaced by a cascaded system of $[H_1]$ and $[H_2]$ as in Fig. 3 and $\underline{f}$ can be estimated by multiplying $[H_2]^{-1}$, followed by $[H_1]^{-1}$, to $\underline{g}$ with the hope that $[H_1]$ would be so simplified in structure or so small in size compared to [H] that the extra effort on $[H_2]^{-1}$ would be warranted. For this purpose $[H_2]^{-1}$ is called preprocessing. Examples of preprocessing are: range alignment, presumming, de-chirping, and motion compensation. Some of them will be discussed in the following sections.

RANGE CURVATURE AND RANGE BIN ALIGNMENT

In general, the radar return of the signal pulse from the target provides the range information while the history of the returns along some range bin provide azimuthal information. These two sources of information could be coupled such that a separable or even separate processing would not be adequate to recover the information to the extent of accuracy one pursues. There are two major sources of non-separability in the radar system: range walking and data misalignment. We now describe the phenomena and propose methods to avoid or correct them.

A. Range Curvature

A single radar pulse return contains the information about the surfaces or lines whose points are equi-distant from the radar transmitter. These surfaces or lines can be resolved by the timing (for short pulse) or range compression (for long duration linear FM-like pulse) techniques. Since the range direction has been compressed and resolved in our source data, the simplest way to resolve the azimuth would be to do one-dimensional processing along cross range direction. This requires that each particular point have contribution to only those range bins which are aligned for azimuthal processing. Such is the case for low or medium resolution SAR imaging with aligned returns. As the resolution requirement becomes greater and greater recently, one is usually forced to reduce the range bin width and/or to increase the azimuthal interval over which the data are to be processed coherently. Both of these would eventually create range curvature problems since the surfaces of constant range as mapped on the target move further away as the relative motion between the radar and the target continues [4,5].

B. Range Alignment

In addition to the range curvature, there is another problem which hinders the separability of the processing: range misalignment. As described before, azimuthal processing operates upon the returns which came from target points at equal range. Thus precise timing or other schemes on returns of individual pulses to insure correct range bin alignment is of ultimate importance to warrant separable processing.

In the data of our radar system, range tracking is provided by a Poly/ Kalman estimator which tries to lock the first strong peak of each pulse return onto a specific range bin. For example, if the point on the target closest to the radar is the wing tip, then the wing tip returns of different pulses hopefully will be locked in the same range bins. Because of scintillation of the reflectivities, this range locking method is not always reliable and misalignment occurs from time to time.

MOTION COMPENSATION

As described earlier, there are two kinds of phase variations induced by motion of the target: motion of the target center relative to the radar and that of the different target points relative to the target center as viewed from the radar. Only the latter contributes to the imaging ability of the radar. It can also be shown that the relation between the latter phase variation and the target reflectivity is a simple Fourier transformation in the azimuthal direction. Thus, a motion compensation of $[H_2]^{-1}$ which removes the effect of the motion of the target center is highly desirable.

Since the trajectory of a single target point is very similar to that of the target center, the returns from that point, if available, can as well be used as a reference to compensate for the target center motion. In fact, this is equivalent to considering this target point as the rotation center of the target. The phases of this reference point, as a function of azimuthal signatures, can then be subtracted from those of all the range bins at the corresponding signatures. Care should be exercised to assure two things: first, the size of the reference point must be small enough. This is because the size of the reference point decides the best possible azimuthal resolution. Second, for each signature, the reference range bin must correspond to the reference point if the advantage of a fast separable processing is to be taken. This requires range alignment as described before.

PRESUMMING

The purpose of presumming is to remove the factor of oversampling in the azimuthal direction. Usually the radar imaging system is oversampled in the azimuth direction because of a too high PRF. In the case of terrain imaging, oversampling is sometimes a result of not processing the whole antenna illumination pattern along the azimuth direction. In that case, the pattern width utilized or coherently processed determines the resolution of the image. In the case of aircraft imaging, the situation is different. Here the azimuthal width of the aircraft is so small that we would always like to make full use of the maximum width of the effective radar illumination pattern, which is the azimuthal length of the aircraft itself. Under this condition the PRF required is decided by the azimuth dimension on the aircraft and the azimuth resolution is decided by the signatures coherently processed. Thus, assuming other parameters fixed, a larger aircraft would require a higher minimum PRF to insure that no aliasing will occur in the final images. Also, since the effective antenna illumination (i.e., overall aircraft azimuthal length) is independent of the wavelength, λ , the minimum PRF or the resolution in the aircraft-imaging case would be functions of λ. This is in contrast to the ground terrain imaging cases where the full antenna illumination pattern width, which is proportional to λ, is to be fully used so that the resultant resolution is independent of the because of a cancelling effect. [1,2]

EXPERIMENTAL RESULTS - FIRST INTERVAL

The mode of the radar system in which our source data was acquired was a wide band high range resolution mode. The transmitted pulse was a linear FM and the pulse returns have been compressed using matched filtering techniques in the radar receiver.

A condensed overall view of magnitude part of the first interval data is shown in Fig. 4 in which each row corresponds to the logarithm of the magnitude of the return from a single pulse. Only every 16th signature is shown in this figure. Recalling that this interval represents the radar returns when the target aircraft was flying toward a broadside position (Fig. 1), we presume that the first high-intensity bins correspond to the left wing tip and the next distinct strong returns are from the fuselage and nose. Note that the radar is to the left of this figure.

Then it can be perceived from Fig. 4 that the fuselage is at a greater and greater distance away from the wing tip along the range direction, as a result of closing-to-broadside during flight. It is also observed that while most portions of Fig. 4 seem pretty well range-aligned, other portions do need re-alignment before a separable processing can be implemented.

To present the data in detail all of the first 512 signatures are displayed in Fig. 5. The phase image (Fig. 5b) indicates clearly that the target points probably lie in range bin number 50 to 200, where a strong structure of phase relationships appear as a result of the coherent radar pulsing. This is also shown in the log magnitude picture Fig. 5a, although with less clarity. There is a transient region where the strength of the returns decreases gradually with the range or time. This phenomena is conjectured to be a result of multiple reflections on the target which took more time before re-radiating to the radar receiver.

To investigate further the behavior of the returns, only the regions of strong signal returns are kept and a sequence of 4096 signatures is shown in Fig. 6 with both log magnitude and the corresponding phase. Observe the quadratic-like phases along the flight direction due to the flight geometry, as analyzed earlier in this report.

Since the radar receiver has range compressed the signal returns we will need only to perform some azimuthal processing. For convenience we transpose the data so that the horizontal direction now denotes the signature or azimuth direction.

Another motion compensation scheme somewhat independent of the flight geometry and very simple in implementation is to use the signal returns from a reference point to estimate the history of the flight range trajectory. This single point can be thought of as the center of rotation of the target and its phases can be subtracted from those of all range bins to leave only the phase histories of all target points relative to this reference point. This was, in fact, the technique used in subsequent imaging.

Figure 7 is a series of processed aircraft images using the above reference point scheme. Consecutive pictures represent abutting 2048 signatures or 20-second flight time each. The images are linearly interpolated in azimuth to give the same range and azimuthal bin width such that the images are correctly scaled. Visually Fig. 7d is the best probably due to best range alignment of the data in that time interval.

EXPERIMENTAL RESULTS - SECOND INTERVAL

The first 8000 signatures of the second interval source data which were taken when the airplane was making a standard left turn are shown in Fig. 8 and Fig. 9. Unlike the straight flight, the phase plot here has a changing azimuthal structure due to the turning motion of the target, which creates complicated range and Doppler histories. In addition, there are several occasions when the range bins are seriously out of alignment. The overall view of Fig. 8 shows the changes of relative positions of nose, fuselage and wing tip due to the turn. A portion of data was taken when the airplane was nose into the radar and a series of resultant images are shown in Fig. 10 using the reference-point technique as a phase compensator. In this case the nose tip serves as a very good reference point as shown by the degree of sharpness of the nose in these images.

The spread patterns close to the nose are due to the aircraft radar which was constantly scanning during the flight, presenting an object of changing reflectivity and violating the assumption that the target was a rigid body in the processing technique.

RANGE RE-ALIGNMENT RESULTS

As is evident from Figs. 8 and 9 the radar breaks range lock quite often during the turn of the target aircraft. This is to be expected as different scatterers from the aircraft dominate the leading return of the radar reflection. Naturally when the radar breaks lock, one would not expect to be able to image without re-alignment processing. An earlier section presented a theoretical discussion on such re-alignment procedures and this section will present some experimental results.

Figure 11(a) presents a typical break in the range lock for a sequence of 512 signatures during the turning portion of the flight. The first returns, which are not very distinct in the first 50 and last 200 signatures, are from the nose tip. The second strong returns are from the left wingtip. Reflectivity of the nose tip scintillated and the wingtip returns were taken for the nose from time to time. Fig. 11(b) is the image of the data of Fig. 11(a). As one would expect, the image looks blurred due to the mixture of the returns from the wingtip and nose after the azimuth processing. However, general orientation of the fuselage is resolved.

A realignment scheme of correlating the magnitude of the returns as described in an earlier section was applied on Fig. 11(a) to become Fig. 11(c). While the scheme works quite well in the neighboring signatures, exponential weights have been applied to the previous aligned data for the correlation reference to insure global alignment.

Fig. 11(d) shows the target image obtained from the religned data. Very much like Fig. 10 this image shows clearly the orientation and the wingtips of the aircraft. However greater structure is now evident as would be expected from properly realigned data.

REFERENCES

1. Chen, C.C. and Andrews, H.C., "Multifrequency Imaging of Radar Turntable Data," submitted for publication in *IEEE Transactions on Aerospace and Electronic Systems*.

2. Harger, R.O., *Synthetic Aperture Radar Systems*, Academic Press, 1970.

3. Andrews, H.C. and Hunt, B.R. *Digital Image Restoration*, Prentice-Hall, 1977.

4. Leith, E.N., "Complex Spatial Filters for Image Deconvolution," *Proceedings of the IEEE*, Vol. 65, No. 1, January 1977.

5. Leith, E.N., "Range-Azimuth-Coupling Aberrations in Pulse Scanned Imaging Systems," *JOSA*, Vol. 63, No. 2, February 1973.

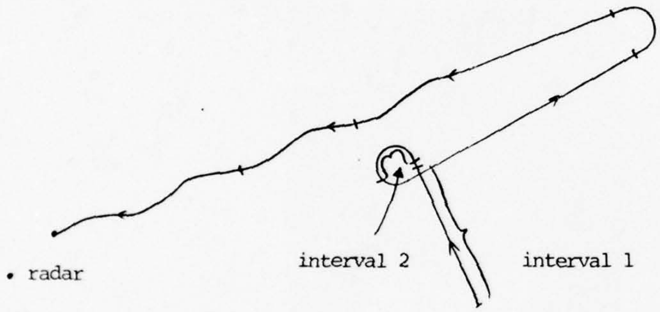
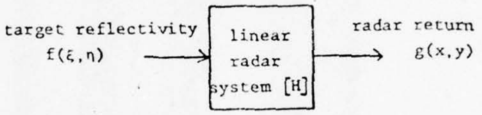


Fig. 1. Overall Flight Path of Target Data.



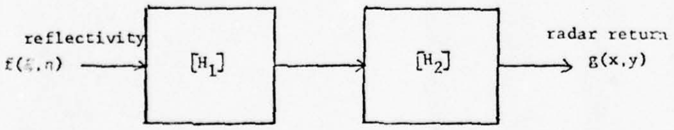
$g = [H]\ f$

[H]: Point spread function matrix (PSF)

$[H][H]^t$: correlation matrix

- determines systems degrees of freedom (DOF).

Fig. 2. Linear Radar Imaging System.



$[H] = [H_2]\ [H_1]$

$[H^{-1}] = [H_1]^{-1}[H_2]^{-1}$

$[H_2]^{-1}$: Preprocessing

- Range alignment
- Range Walking
- Motion Compensation
- Presuming

$[H_1]^{-1}$: Ideally a Fourier transform

Fig. 3. Decomposition of [H].

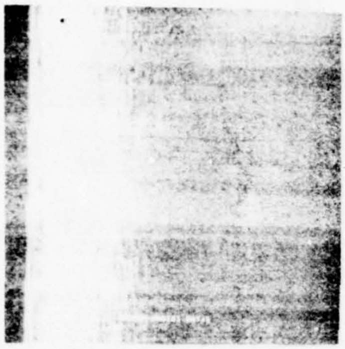
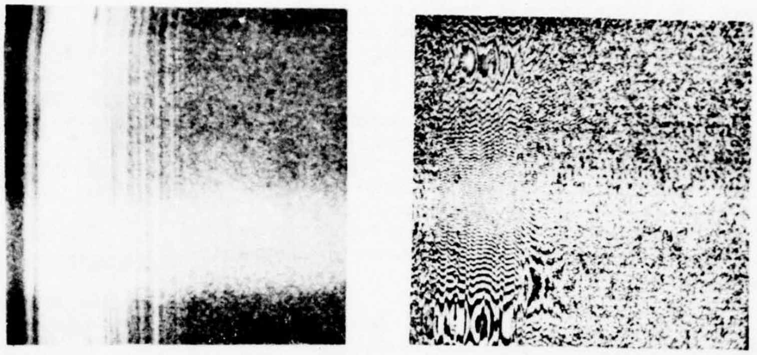


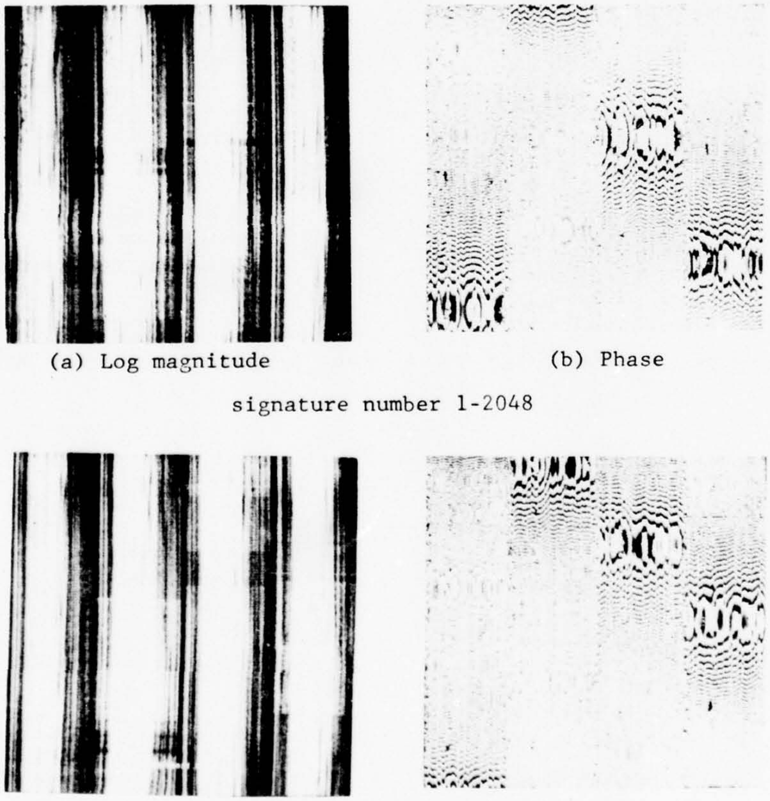
Fig. 4. Overall view of first interval data; log magnitude of every 16th pulse return.



(a) Log magnitude

(b) Phase

Fig. 5. First 512 Signatures of First Interval Data.



(a) Log magnitude

(b) Phase

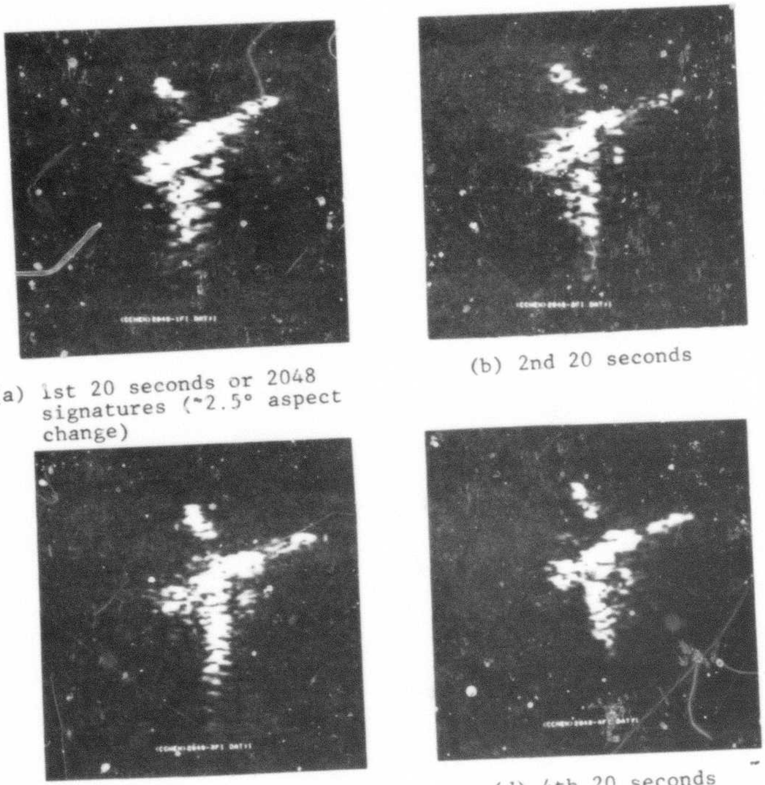
signature number 1-2048

(c) Log magnitude

(d) Phase

signature number 2049-4096

Fig. 6. First Interval Data With 128 Range Binds Stacked Side by Side.



(a) 1st 20 seconds or 2048 signatures (~2.5° aspect change)

(b) 2nd 20 seconds

(c) 3rd 20 seconds

(d) 4th 20 seconds

Fig. 7. Aircraft Radar Images with Abutting 20 Second Coherence Time.

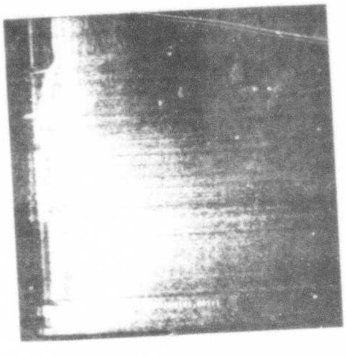
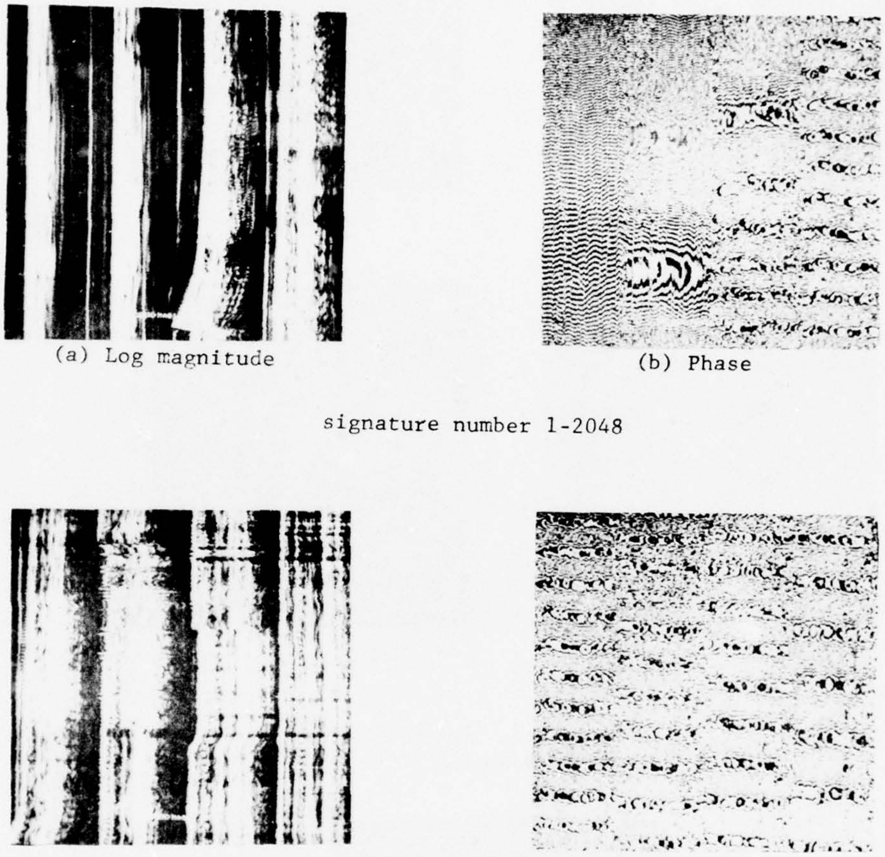


Fig. 8. Overall View of Second Interval Data; Log Magnitude of Every 16th Pulse Return.



(a) Log magnitude

(b) Phase

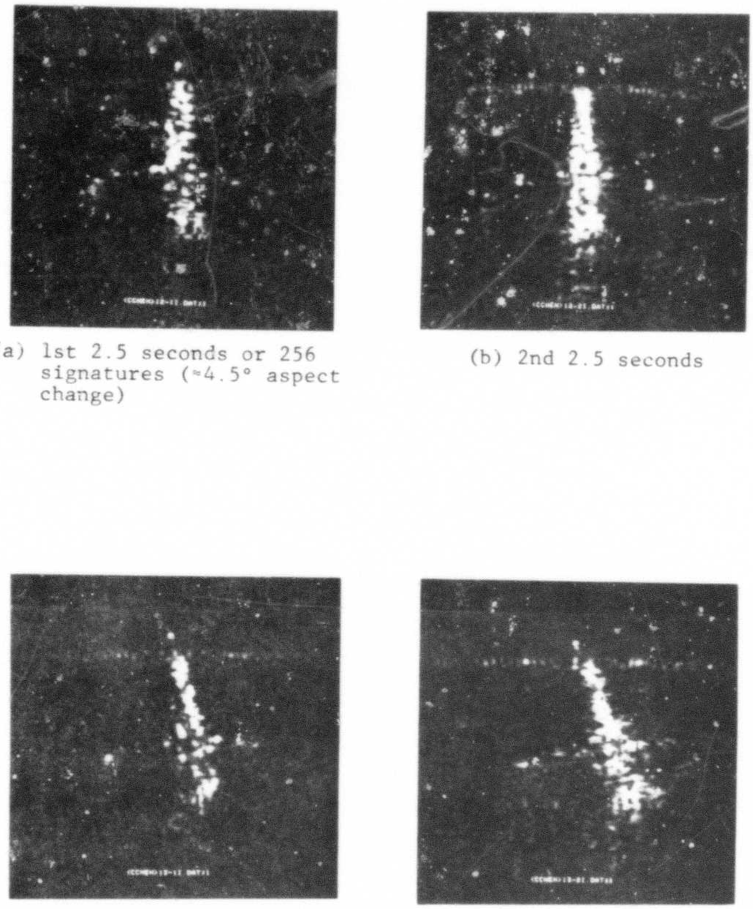
signature number 1-2048

(c) Log magnitude

(d) Phase

signature number 2049-4096

Fig. 9. Second Interval Data With 128 Range Bins Side by Side.



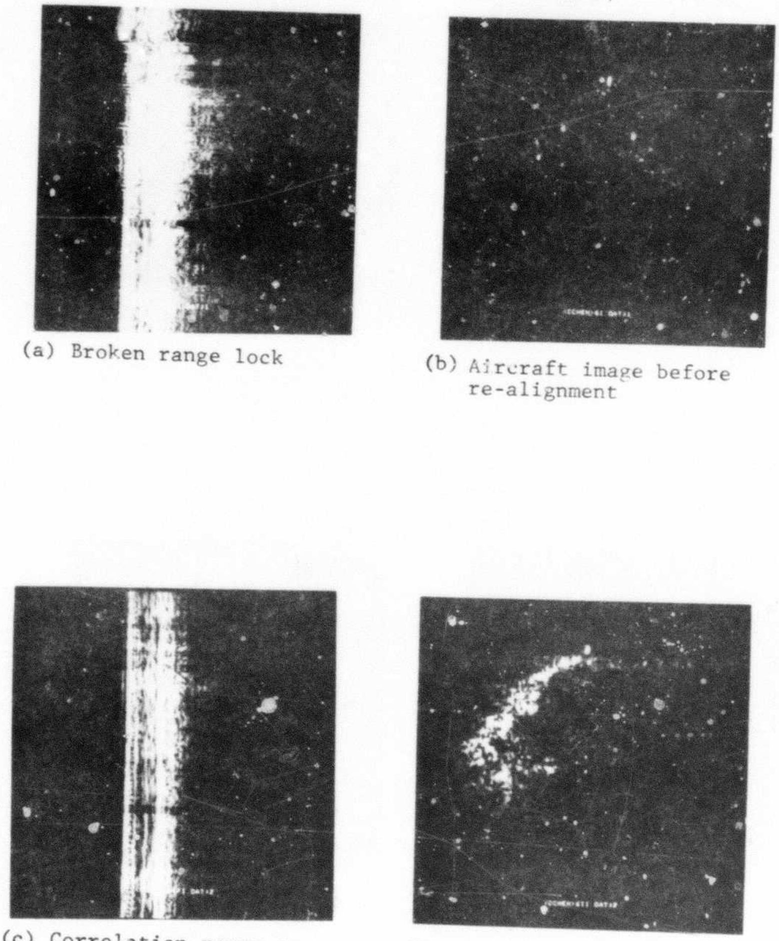
(a) 1st 2.5 seconds or 256 signatures (≈4.5° aspect change)

(b) 2nd 2.5 seconds

(c) 3rd 2.5 seconds

(d) 4th 2.5 seconds

Fig. 10. Aircraft radar images with abutting 2.5 second coherence times.



(a) Broken range lock

(b) Aircraft image before re-alignment

(c) Correlation range re-alignment

(d) Aircraft image after re-alignment

Fig. 11. Range re-alignment.

SESSION IV

SYSTEMS

EFFECTIVE TRANSMISSION OF RASTER IMAGES

Kenneth R. Sloan, Jr.

Department of Computer Science
University of Rochester
Rochester, New York 14627

ABSTRACT

The transmission of high resolution raster images over low-bandwidth communication lines requires a great amount of time. User interaction in such a transmission environment can be frustrating. The problem can be eased somewhat by transmitting a series of low resolution approximations, which converge to the final image. A method of computing such a series of images which requires no transmission overhead and only a small amount of local computation is presented.

INTRODUCTION

Raster graphics display devices are capable of reproducing very complex images. Unfortunately, they are often connected to the source of those images, a large mainframe computer, by low-bandwidth data links. This makes it difficult to interact effectively with the display when it is being used to display the images for which it was made (often full-color, typically 512*512 picture elements (pixels)). Transmitting such an image over a 1200 baud line can take half an hour, or longer. If it is being displayed on a line-by-line basis, then it may be 15 or 20 minutes before the user has any notion of what the final picture will be like.

This problem can be alleviated somewhat by sending, and displaying, a series of images which converge to the final, full resolution picture. Successive images are refinements of earlier images, and approximations to the original image. The primary advantage of such a scheme is that global structure in the image becomes apparent very early in the display process, allowing the user to begin to examine the picture, and even interrupt the display when satisfied with the approximation. The disadvantages lie in (possibly) increased storage or computation costs.

PYRAMID DATA STRUCTURES

A pyramid data structure consists of several levels, numbered 0-L, where each level is a 2-dimensional raster image. Level L is the most detailed (finest resolution) image; the others are derived from it, and are approximations to it. The value of a pixel in level k is a function of the values of the pixels in an MxN window in level k+1. Thus, the relevant parameters of a pyramid data structure are:

a) X,Y : the dimensions of Level L,
b) M,N : the dimensions of the reduction window,
c) R : the reduction rule.

Usually, the reduction window and the original image are square (M=N, X=Y=(M**L)), but these conventions can be relaxed, at some cost in computational complexity. The reduction rule can be any reasonable function of the pixels in the window (e.g., Min, Max, Mean, Median, Mode, Sum, Selection, or their extensions for handling colored pixels).

NAIVE METHOD

Assuming that a pyramid data structure has been built, there is a straight-forward display technique which depends only on the ability of the local processor to paint rectangular regions on the screen (or in a frame buffer). The pyramid is simply transmitted "top-down". Each level is sent in the usual raster scan order, and used to overpaint the existing image (Figure 1). First, level 0 (1x1) is painted as a single block, covering the entire screen. Then level 1 (MxN) is sent and displayed, again filling the entire screen. Successive levels, requiring ever increasing amounts of time to transmit and display, serve to continually refine the details of the image on the screen (see Figures 2 and 3).

This method can be used to display any pyramid data structure, regardless of the choice of reduction window size and reduction rule. However, since each level is sent in its entirety, all of the effort devoted to sending levels 0 through (L-1) is "wasted" when level L completely overwrites it. When the reduction window is 2x2, this means a 33.3% increase in transmission time for the full resolution picture. Also, there must be a small amount of local (to the display) computation and state, which interprets the sequence of pixel values and keeps track of such things as the current level, the position within the current raster scan, and the size of the rectangles to be painted. A small amount of preliminary information may need to be transmitted in order to initialize this local computation. This transmission overhead is negligible, however.

OMIT REDUNDANT PIXELS

The Naive Method uses knowledge, common to Sender and Receiver, about the breadth-first scan of the pyramid data structure. If the Receiver also knows the reduction rule R which was used to grow the pyramid, then we can avoid sending certain "redundant" pixels. In general, this will work for any reduction rule which allows the derivation of the Value of a single Son pixel, given the Values of the Father and the remaining Sons. In particular, if the reduction rule is Selection (Value of Father = Value of Son[x',y']), then not only can we avoid sending Son[x',y'], but we do not even have to derive its value! When the other Sons are transmitted and painted on the screen, Son[x',y'] is already correctly painted on the screen. The area corresponding to Son[x',y'] was painted when the Father was painted, and does not need to be repainted. The point is that both the Sender and the Receiver can know this.

As with the usual row-by-row raster scan, we must transmit X*Y pixels. This means that there is absolutely no transmission overhead, compared with a row-by-row painting of level L. The advantages of early presentation to the user of a complete, albeit low resolution, image are obtained at the price of a small amount of computational overhead. Also, since the Receiver need not refer to previously sent pixels in order to derive the value of the "missing" pixels, only display operations are required of the Receiver (Figure 4).

The values transmitted correspond exactly to the values of the pixels at level L. The order in which they are sent is the only difference between this method and the traditional row-by-row raster scan. Just as the Receiver must understand the ordering of the usual raster scan, the Receiver for this method must understand, and properly interpret, this ordering. If the time to write a large rectangular area on the screen (or in a frame buffer) is "free" compared with the transmission time, then this method is "free".

INTERACTIVE DETAILING

Both of the above methods can be modified to allow the observer to direct the successive refinement process. Once the entire image has been painted to some minimum resolution, the user may interrupt the transmission of the image and indicate an area to be refined further. The refinement process is then limited to that area of the image. This will prevent the transmission of information about areas of the image which are uninteresting to the user, and allow much faster refinement of the important details.

TRANSFORM METHODS

The two methods discussed above yield a "series" representation of the image, and have the "prefix property". That is, truncating the series at any point gives an approximation to the original image. There are, of course, other representations with this property. Two which have been used extensively in image processing are the Fourier and Hadamard transforms [Andrews, 1970]. The primary difficulty with such methods is the amount of computation required to turn the representation into a visible image. If this is to be done only once, after complete transmission of the (truncated) transform, then this might not be a serious objection. However, it is not immediately clear how to extend these methods to interactive detailing in the spatial domain.

The methods described have the additional property that they are well matched to the display capabilities of available raster graphics equipment. For example, painting a rectangular block is essentially free on many display devices. Since the display equipment provides the transform inversion, this means that rapid, repeated, incremental conversion of the series representation into a viewable image is feasible.

CONCLUSION

The widespread use of high resolution raster graphics displays will require effective use of low bandwidth communication lines. Methods of transmitting raster images which provide early recognition of gross features and which are well matched to available display devices are examples of such effective use of bandwidth. The use of these methods is by no means restricted to display applications. They are suitable for any situation in which the Receiver can make use of a low-resolution image, especially when the required resolution is not known a priori.

REFERENCES

Andrews, H.C. Computer Techniques in Image Processing. Academic Press, 1970.

Klinger, A. and C.R. Dyer. "Experiments on picture representations using regular decomposition." Computer Graphics and Image Processing 5, 1 (1976), 68-105.

Sloan,Jr., K.R. and S.L. Tanimoto. "Progressive Refinement of Raster Images", TR 39, Computer Science Department, University of Rochester, October 1978.

Tanimoto, S.L. "A pyramid model for binary picture complexity." Proceedings IEEE Computer Society Conference on Pattern Recognition and Image Processing, Troy, NY (June 1977), 25-28.

Tanimoto, S.L. and T. Pavlidis. "A hierarchical data structure for picture processing." Computer Graphics and Image Processing 4, 2 (1975), 104-119.

Warnock, J.E. "A hidden-surface algorithm for computer-generated half-tone pictures." TR 4-15, Computer Science Department, University of Utah (1969).

NAIVE SENDER

```
begin "send image"
 for level := 0 step 1 until L
  do begin "send level"
      for y := 0 step 1 until (N**level)-1
       do begin "send scan line"
           for x := 0 step 1 until (M**level)-1
            do Send(Pyramid[level,x,y])
           end  "send scan line"
      end "send level"
 end "send image"
```

NAIVE RECEIVER

```
begin "receive image"
 for level := 0 step 1 until L
  do begin "receive level"
      for y := 0 step 1 until (N**level)-1
       do begin "receive scan line"
           for x := 0 step 1 until (M**level)-1
            do begin "receive pixel"
                Receive(pixel);
                x1 :=    x    * ScreenMaxX / (M**level);
                x2 :=  (x+1)  * ScreenMaxX / (M**level)-1;
                y1 :=    y    * ScreenMaxY / (N**level);
                y2 :=  (y+1)  * ScreenMaxY / (N**level)-1;
                SetColor(pixel);
                PaintRectangle(x1,y1,x2,y2);
                end  "receive pixel"
            end  "receive scan line"
      end  "receive level"
 end   "receive image
```

Figure 1

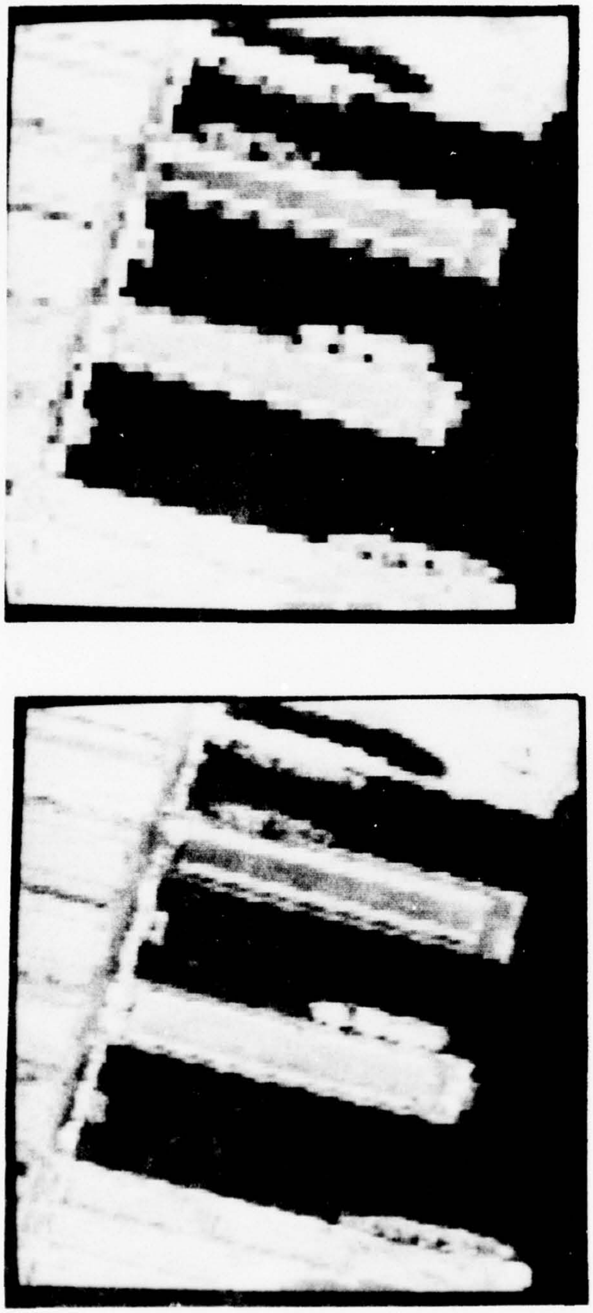


Figure 2

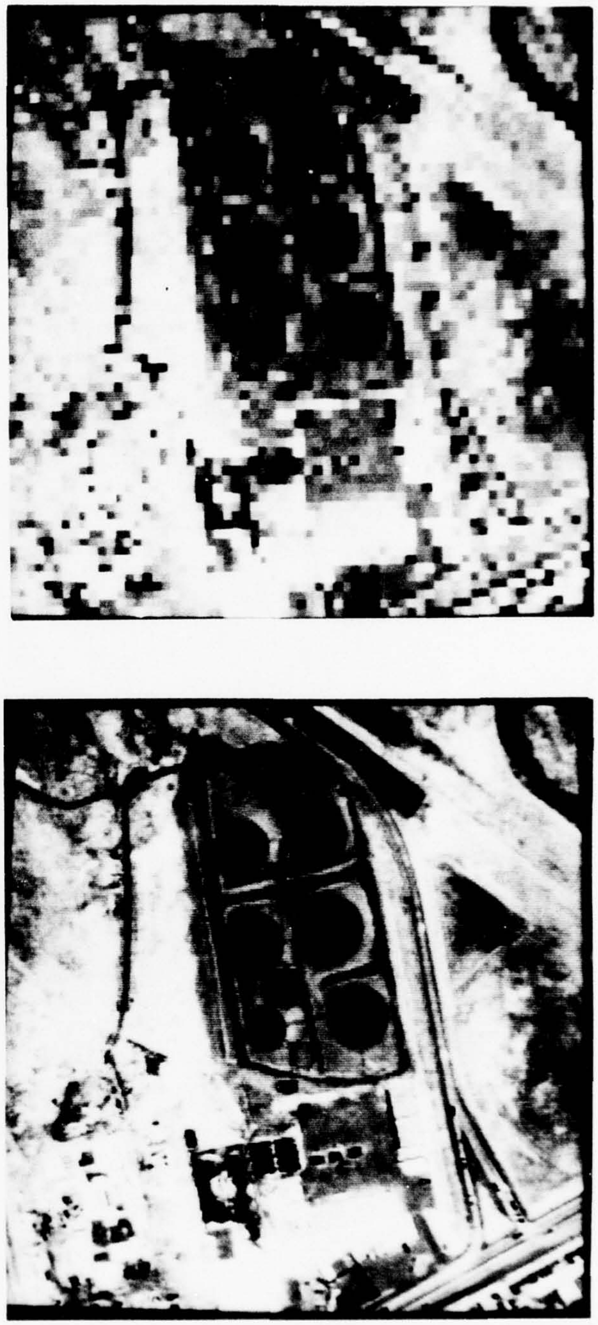


Figure 3

OMIT REDUNDANT PIXELS (SELECTION) SENDER

```
begin "send image"
 for level := 0 step 1 until L
  do begin "send level"
      for y := 0 step 1 until (N**level)-1
       do begin "send scan line"
           for x := 0 step 1 until (M**level)-1
            do begin "send pixel"
                if ((y MOD N) NEQ 0) OR ((x MOD N) NEQ 0)
                    OR (level = 0)
                 then Send(Pyramid[level,x,y])
                end  "send pixel"
           end  "send scan line"
      end  "send level"
 end  "send image
```

OMIT REDUNDANT PIXELS (SELECTION) RECEIVER

```
begin "receive image"
 for level := 0 step 1 until L
  do begin "receive level"
      for y := 0 step 1 until (N**level)-1
       do begin "receive scan line"
           for x := 0 step 1 until (M**level)-1
            do begin "receive pixel"
                if ((y MOD N) NEQ 0) OR ((x MOD N) NEQ 0))
                    OR (level = 0)
                 then begin "overpaint with son"
                       Receive(pixel);
                       SetColor(pixel);
                       x1 :=   x   * ScreenMaxX / (M**level);
                       x2 := (x+1) * ScreenMaxX / (M**level)-1;
                       y1 :=   y   * ScreenMaxY / (N**level);
                       y2 := (y+1) * ScreenMaxY / (N**level)-1;
                       PaintRectangle(x1,y1,x2,y2)
                       end "overpaint with son"
                end  "receive pixel"
           end  "receive scan line"
      end  "receive level"
 end  "receive image"
```

Figure 4

PROGRESS REPORT ON A MODEL-BASED VISION SYSTEM

Rodney A. Brooks, Russell Greiner and Thomas O. Binford

Artificial Intelligence Laboratory, Computer Science Department
Stanford University, Stanford, California 94305

Abstract

We report on development progress with a model-based vision system called ACRONYM. The system is being built with airfields, oiltanks, aircraft, buildings, and vehicles as examples for interpretation and measurement. It uses shape and symbolic models in a more powerful way than other approaches and is expected to lead to PI systems capable of monitoring, measuring and counting. The user is able to model objects and their spatial relations in terms of high level spatial constraints. Goal reduction methods are used to infer quasi-invariant observable features of an object. Instances of such objects are found in an image by a cost efficient matcher which employs a coarse to fine strategy. Included is a facility which attempts to justify the absence of some required feature by building hypotheses which later must themselves be validated.

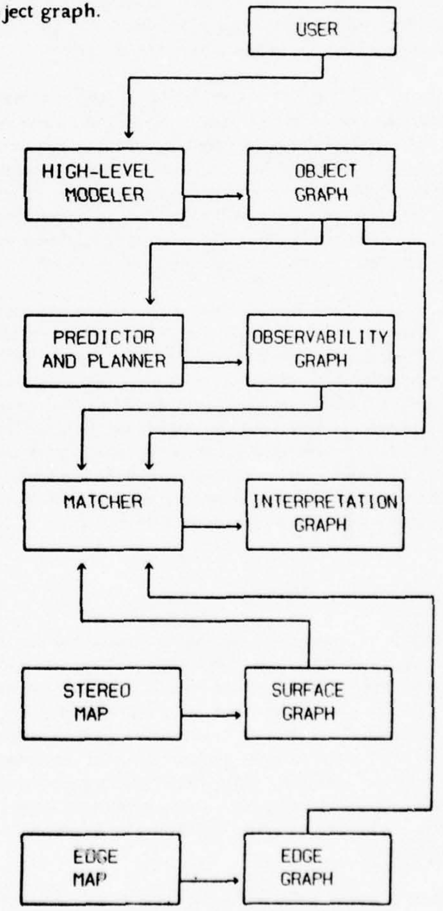
Introduction

In Brooks, Greiner and Binford [1978] we described the early design and implementation of a model-based vision system, called ACRONYM. We regard the system as a vehicle for research into the problems of identifying objects based on generic descriptions, and of providing tools for users to specify vision tasks in a natural way. An objective is to implement the system in ways that are robust and generalizable. In a typical scenario, a photointerpreter will give a brief symbolic description of a typical airfield, and describe some specific airfields. He will show some examples of airfields, from which both specific and generic properties will be inferred. We are also exploring the integration of many techniques developed here and elsewhere in vision and modeling projects.

Figure 1 is a schematic of the logical modules of the ACRONYM system and its operating environment. Images to be processed by ACRONYM are preprocessed by other systems. We will use Arnold's [1978] edge matching stereo system to provide a depth map of the scene. Nevatia and Babu [1978] have developed techniques which will be useful for extracting shape descriptions of regions within the images to produce the picture graph.

The ACRONYM system itself has three main modules: the high level modeler whose output is an Object Graph; the predictor and planner whose output is an Observability Graph; and the matcher whose output is an Interpretation Graph.

The user gives generic descriptions of objects in a high level modeling language. The representations of objects is usually very compact; they are segmented into volume elements known as *generalized cones* (see Binford [1971] or Agin and Binford [1973]). The key design requirement was that the primitives in the volume representations must aid in generic description of parts of objects. Generalized cones were originally designed to satisfy this requirement. The volume elements and their spatial, and functional relations are combined to form the Object graph.



Model-Based PI System
Figure 1.

The predictor and planner module has two main functions. It produces and displays perspective projections of the appearance of the modeled three dimensional objects providing essential feedback for the user. Its main function is to produce the Observability Graph. The Observabilty Graph is a symbolic summary of the modeled objects in terms of their quasi-invariant observable features and relations between them. Observables are those features and relations which are detectable, i.e. that are easily found by operators; they are expected to have reasonable contrast and be large enough to find. Quasi-invariants are those features which remain nearly invariant over a large range of viewing angles. The reasoning about the appearance of objects is carried out in the three dimensional domain. This allows use of knowledge about the three dimensional spatial inter-relations of cones to be used in deciding the shapes which will appear in the two dimensional image. So far the production of graphics and the Observabilty Graph have been carried out almost independently. It is expected that they will become more intertwined as both tasks will be able to share portions of specialized knowledge.

The third module of ACRONYM is the matcher. Matching is carried out by a relaxation process. The conditions which go into detailed verification vary enormously in their cost and effectiveness. A general structuring of the matching process into coarse and detailed phases reflects an ordering of priorities. Local shape elements give clues to the detailed matching. The matcher integrates segmentation with identification. Thus we will not need complete or perfect segmentation of the image.

Since our last report much of the original code has been rewritten in terms of a data manipulation language written for the purpose. While not as powerful as some others (e.g. the FRL system of Roberts and Goldstein [1977]) our data language tackles the same problems of providing defaults, constraints, procedural attachment, inheritance and associative retrieval for data structures. Our system is a preprocessor which produces very efficient LISP code which is then compiled. Thus while retaining efficiency, our new modules are easily modified to accommodate increasingly more general data structures.

Modeling

Our goal is to implement modeleing capabilities for a subclass of cones adequate for susbsequent preicion and matching. At the time of our previous report (Brooks, Greiner and Binford [1978]), only generalized cones represented in the first line of the table of fig. 2 were implemented, i.e. only cones with straight spines, cross sections perpendicular to the spine and constant or linear sweeping rules. Recall that a generalized cone describes a volume by sweeping a cross section area along a spine (some curve in space) while deforming it according to some sweeping rule.

The letters C and L refer to constant and linear sweeping rules respectively. The presence of such a letter in a box indicates that ACRONYM can handle generalized cones with that type of sweeping rule, for the given spine and cross section characteristics. A constant sweeping rule sweeps out the cross section without change. A linear sweeping rule scales the cross section linearly with the distance swept along the spine.

The introduction of circular spines allows better modeling of taxiways, and curved roadways in general. The cross section used for taxiways, and indeed all ground surface markings, is a rectangle of zero height. However, analytic solutions were found for the more general problem of what portions of a generalized cone with circular spine and convex polygonal cross section are visible from a given camera position. See fig. 4 for an example.

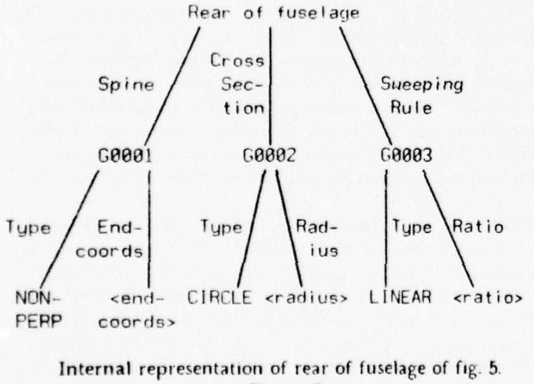
Spine Type/Angle between spine and cross section. / Types of segments needed for outline of cross section.	Straight	Circular
Straight/perpendicular	C L	C L
Circular/perpendicular	C	
Straight/non-perpendicular	C L	C L

Taxonomy of generalized cones.
Figure 2.

A new subclass of generalized cones, where the cross section is kept at some non-normal angle while being swept along the spine, were introduced as a result of attempts to model an airplane (a Lockheed L1011, see fig. 5). Previously we had required that the cross section be kept normal to the spine while being swept along. The rear section of the fuselage in fig. 5 has been modeled by a cone with a linearly decreasing sweeping rule and a circular cross section, held at a fixed angle while being swept along a straight spine, inclined to the main axis of the fuselage. The internal representation for such a cone does not explicitly contain either the length of the straight spine, or the angle between the cross section and the spine. Rather, it has the form shown in fig. 3. As previously reported we use a local coordinate system for each generalized cone. The coordinate system is centered about the specified cross section rather than about the spine. Thus for these types of cones the specification is in terms of the two ends rather than the spine. The specified cross section lies in the y-z plane with the spine intersecting it at the origin. We previously required that the tangent to the spine at the origin be the x-axis. This was to ensure that the cross section and spine specifications were perpendicular there. For "NON-PERP" cones we merely require that the coordinates specified for the other end of the spine lie in the non-negative x half space.

The wings and stabilizers are also best modeled with cones with non-perpendicular cross section. The streamlines underlying the physical design lie parallel to the main axis of the fuselage, whereas the natural spine follows the sweep of the wing which is not perpendicular to the fuselage. The wings can be specified in terms of their cross section at the fuselage, and the position of the wing tip.

With the introduction of these new classes of cones, ACRONYM can handle a much larger class of surfaces than other volume based modeling systems. It implements a larger subclass of generalized cones than the previous system of Miyamoto and Binford [1975]. Generalized cones as initially proposed (Binford [1971]) were very general. Most other modeling systems are restricted to planar surfaces (e.g. Baumgart [1974], Miyamoto and Binford [1975], Grossman [1975]) or perhaps planar and cylindrical surfaces (e.g. Braid [1973], Voelcker [1974]). Agin [1972] used cones which always had circular cross sections, spines made up of straight line segments and linear sweeping rules. We have determined fast analytic solutions for the appearance of our larger class of surfaces. These will be useful for both producing Observabilty Graphs of generic objects from generic views and for back solving for objects from the image. Also, since the number of surfaces is greatly reduced by having these better approximations to an object being modeled, we believe that we will be able to achieve much faster display programs. These analytic solutions have been implemented in our graphics modules. We have not yet implemented a full hidden surface elimination algorithm - we eliminate only back surfaces and other surfaces occluded by part of the same generalized cone which generates them. We have found a number of analytic solutions necessary to extend some of the classical hidden surface algorithms for planar surfaces (see Sutherland, Sproull and Schumaker [1973]) to our more general classes of surfaces. Our analytic solutions will appear elsewhere at a later date. We intend to find solutions for a broader subclass of cones. We also intend to implement two dimensional generalized cones as one representation for surfaces Non-cone representations are included where convenient.



Internal representation of rear of fuselage of fig. 5.
Figure 3.

Producing the Observability Graph

Producing the Observability Graph requires reasoning about the spatial relationships and shapes of three dimensional generalized cones. This is an expert task, and requires explicit expert knowledge. At two extremes this knowledge can be imbedded in a program as its control structure, or it can be given to some program as data. The MYCIN group at Stanford (see Davis, Buchanan and Shortliffe [1975]) have developed a number of systems close to the latter end of this spectrum, which have achieved performance in restricted domains, close to or better than that of human experts. The systems constructed have included experts on secondary bacterial infections, pulmonary function, molecular genetics, and an expert consultant on how to use a large and expensive to run, structural analysis computer program MARC (see Bennett, Creary, Englemore and Melosh [1978]).

These systems have all been rule based and used backward chaining as their main control structure. The rules are small pieces of knowledge, or advice, on how to solve the problem being tackled. They can be paraphrased in English as saying something of the form: *If this list of statements is true, then these other statements are also true.* Typically human experts in the particular field of study, have written down pieces of their knowledge of the subject in the form of these rules. These are then translated into machine readable form. Ideally the expert should not have to worry about the control structure used by the program. Rather he considers each rule as a piece of advice which may be useful in solving problems of this particular type. Representing the problem solving knowledge in this way has a number of advantages. When new knowledge is to be added to the program, no reprogramming need be done, rather a new rule can just be added to the rule base. Davis [1976] has developed the idea of *meta-rules*. The conclusions of these rules affect the way rules are selected in future by the control structure. Since the problem solving knowledge is all declarative rules, these meta-rules can examine other rules, and so be used to reason about the problem solving process, and modify the behaviour appropriately. This would be very hard if the problem solving knowledge was imbedded in programs.

Encouraged by these successes we decided that a backward chaining rule based system would be a useful starting point to investigate the production of the Observability Graph. A particularly attractive feature is the additivity of problem solving knowledge. As we expand the class of generalized cones and spatial relations being used, we can simply add more rules to explain how to handle the new cases. We have implemented a backward chaining control structure and have experimented with a small set of rules, producing small Observability Graphs. The rules we have written are intended for initial experimentation only. We have carried out a few initial experiments. As a result we have begun to investigate the possibilities of extending the control structure somewhat, and have also decided to expend more effort on two dimensional shape descriptors, as described in the previous section. The rest of this section describes more fully the system already implemented and shows an example of how it works.

The rules have *premises* and *actions*. The premises are sentences about the Object Graph. When a rule is invoked the truth of these sentences is checked. If they are all true the actions are executed. This is a simple programming language. Actions might add information to the Object Graph, construct parts of the Observability Graph, and eventually make changes to the state of the control structure, to affect the choice of rules during future processing. The backward chaining mechanism proceeds as follows. A special rule is invoked, whose only action is to conclude that backward chaining has been completed. Its premises are that certain subtasks have been completed. To check the validity of a given premise, the system looks in an associative data base for other rules whose action list includes one, which with the correct bindings to variables, might assert the premise. The backward chaining mechanism is called recursively to check that the premises of the new rule are satisfied. If so the actions of the rule are executed, the assertion is made and the premise of the original rule is passed as true. Many premises of rules are simple checks against the Object Graph, and so the recursion is halted eventually.

Figure 6 is an excerpt from a trace of the backward chaining system, inferring the observable features of the fuselage of a modeled L1011 airplane. It shows the portion of the computation dealing with the invocation of rule R19. Rule R19 is shown in figure 7. It is an example of a rule with only a single action. It says how to calculate the apparent width of a ribbon in the image, given that the ribbon will appear rectangular in the image, that the cross section of the three dimensional generalized cone producing that ribbon is circular, and that the system has been able to deduce the radius of the cone. The way to calculate that width is to multiply the circular radius by two. The symbol $CONTEXT is a variable which is bound to the cone of current interest at the time the rule is invoked. In this example it is bound to the cone corresponding to the main section of the fuselage of the L1011. The modeler has labeled that cone "FUSELAGE".

Rule R19 has been invoked because some other rule is trying to deduce the rectangular width of a ribbon. It looked in the associative data base for all rules which conclude rectangular widths when their premises are satisfied. Rule R19 was included in the list of such rules, but none of the preceeding rules in that list had all its premises satisfied (in this particular case R19 was actually the first rule in the list). The premises for R19 are checked in order by recursively invoking the backward chaining mechanism. It turns out that it has already been deduced that the FUSELAGE will appear as a rectangular ribbon, so no further rules need be invoked to prove the validity of the first premise. Both rules R12 and R13 are potentially able to prove that a cone has a circular cross section. The premises of both of them involve only simple lookups in the model so no further recursion of the backward chaining will occur here. Rule R13 is tried first, and its premises are not true and so it fails. Next rule R12 is tried. This time the premises are satisfied so the actions are carried out. Rule R12 has two actions. First it concludes that the cross section of the cone is actually circular, and further is able to calculate the radius. The last premise of rule R19 is now checked. It has just been satisfied by the action of rule R12, and so R19 succeeds and records the rectangular width.

Matching

As part of the modeling process, an Observability and Object Graphs are constructed for each object. The Matcher will use these when attempting to locate an instance of this object in an actual scene.

This scene must be transformed from the pixel level in which it is input into a Picture Graph before the Matcher can begin. This preprocessing step begins with an edge detector, which returns an Edge Graph. Each node here refers to a line segment, with parameters which describe its length and straightness, as well as some measure of the intensity gradient across it. The arcs designate spatial relations, such as co-linearity and anti-parallelism. Sysems of co-linear line segments are combined to form one, discontinuous line. Ribbons are constructed by taking anti-parallel pairs of these lines, especially those whose "interior" is relatively uniform in hue or shade. This form of region building is similar to work done by Nevatia and Babu [1978].

These edge pairs, together with data describing the enclosed area, form the nodes of the Ribbon Graph. Its arcs are then determined, based on both the edge-edge relations gleaned from the Edge Graph and properties manifest by the interior of the area. A given edge may have ambiguous interpretations -- i.e. it may have many gaps, or may not fall cleanly into either the straight or circular-arc category. Such edges may be used to border several distinct ribbons. The obvious bookkeeping is done to keep members of such groups mutually exclusive.

In later releases, the low level processer may take into account the specific Observability Graph it is trying to match, and thus be able to perform a goal-directed scan. If so, additional scans will have to be performed during subsequent Matcher phases, when other, more detailed features are sought. As these will be done only when necessary, and over a region now localized by information derived from these earlier passes, this method should prove cost efficient.

An enhanced version of the first part of the matching algorithm, the coarse pass, has been implemented. The overall structure of the algorithm has not changed significantly from the description given in Brooks, Greiner and Binford [1978]. First the nodes of the Observability Graph (which each correspond to some part of the overall object,) are individually matched against the preprocessed input scene. This local information determines which picture nodes might be an instance of this observability node. Each of these observable-part to potential-instance maps is considered a node in the Interpretation Graph. The Observability Arcs are then processed. Two nodes of the Interpretation Graph are joined whenever the implied pair of picture nodes are related in a manner which satisfies some Observability Arc, where this Observability Arc connected the corresponding pair of Observability Nodes. (See figure 8.) Next the graph relations are processed in an analogous manner. The global information derived from these two steps serves to order and possibly prune the candidates for each node. The final step is to form clumps of node to instance mappings by joining together those interpretations which are mutually consistent -- that is, which can be realized simultaneously. Although a given ribbon may have numerous interpretations, any clump may contain at most one of them. The one or more clumps so generated are returned in a best first order. (Finding zero clumps is deemed a failure.)

The remainder of this section will describe several of the behind the scenes changes which have been made to accommodate increasingly more general, and hence more useful, object descriptions. The graph structure used by both the Observability and Object Graphs has been further extended in several ways. Graph Relations were used to describe associations which relate an arbitrary number of nodes. With this tool, it is straight forward to speak, for example, of a connected system of roads, or a cluster of mutually close runways.

Before accepting a candidate Picture Graph Node as an instance of some Observability Node, ON, those arcs with which ON is affiliated may have to pass a series of tests, concerning their nature and number. These requirements may be arbitrarily complex, and may probe other aspects stored in the Interpretation Graph. For example, the user may insist that a ribbon in the input picture qualifies as an Airplane fuselage only if, (in addition to qualifying based on "internal" characteristics,) it intersects with exactly two Wings -- that is, a pair of intersect-arcs each join this candidate with a picture graph node which has qualified as a Wing). The user may additionally insist that the angle of each of these intersections be within a certain range, or that their respective angles, while individually unconstrained, be roughly equal. This same flexibility applies to the requirements which may be placed on the relations associated with each node. One may pass a potential aircraft wing only if it either has exactly one engine pod, or has two, provided there is no pod on the tail of the fuselage. Both of these examples are rather easy to state, but would be quite complicated to implement with the wrong data structure

Each attempt to instantiate an observable feature produces a list of passed and failed tests, as well as a pair of values. The first is a measure of how strong the evidence is that this picture element is indeed an instance of that feature, while the second encodes the evidence to the contrary. This information is used in subsequent stages to order the list of candidates, and to eventually screen out those which appear least viable. (The Mycin Project also employed this method of using both pro and con values -- see Davis, Buchanan and Shortliffe [1973]).

A "Wait and See" philosophy is used throughout this matching process; based on the assumption that additional information available later will provide a better discrimination. As such, everything which is not explicitly excluded will remain a candidate. During the matching, a collection of assumptions and conjectures is maintained. One use of this is to propose a likely interpretation for some not-yet-investigated object found in the picture, based on evidence which is available now, but which will be lost or buried in later phases, when that object is finally queried. The conjecture can also be used to attempt to justify the absence of some expected-but-unfound feature. For example, it makes sense to require that the interior of a runway be fairly uniform in intensity, and that its boundaries be highly visible and unbroken. Imagine that some relatively small and highly reflective object appears on an otherwise acceptable runway, and that this alone keeps both of the above conditions from being met. The matcher will then "conditionally accept" this runway, subject to later verification that that obstructing object is indeed the aircraft it conjectured. It is possible this same potential runway is also a candidate for a highway. Here, finding that object NOT to be a large truck may be just the damning evidence needed to remove that highway interpretation from consideration.

It should be noted much work has to be done to perfect the sort of temporarily unsupported inferences one can and should be able to make. Eventually there will be a battery of stored suggestions, each designed to assist the Observability Graph by making the appropriate conjectures in certain situations. For example, if many parts of the picture appear over-saturated, the global asssumption that the picture contains specular reflection is logical. Based on this, one might test angle relations to verify that the surface properties of other parts are similarly be lost in the glare, and change the expectations for the rest of the picture accordingly.

Acknowledgement

The MACSYMA symbolic mathematics system at MIT was used to aid the calculations of the appearance in perspective projection of the class of curved surfaces described herein. The availability and utility of MACSYMA is gratefully acknowledged. The pictures we work with come courtesy of San Francisco Airport Engineering Department who authorized our use of the pictures and Towill, Inc, San Francisco, who provided the pictures to us. We are especially grateful to Jerry Sample who was extremely helpful in providing us with the pix and very warm to deal with. We are grateful to USC ECL who digitized them for us.

References

Agin, Gerald J. [1972]: *Representation and Description of Curved Objects*, Stanford Artificial Intelligence Laboratory, Memo AIM-173, Oct.

Agin, Gerald J. and Thomas O. Binford [1973]: *Computer Description of Curved Objects*, Proceedings of the Third International Joint Conference on Artificial Intelligence, Stanford, Ca., Aug., PP. 629-640.

Arnold, R. David [1978]: *Local Context in Matching Edges for Stereo Vision*, Proceedings: ARPA Image Understanding Workshop, Cambridge, Mass, May, pp. 65-72.

Bennett, J.S., L.G. Creary, R. Engelmore and R. Melosh [1978] *A Knowledge-based Consultant for Structural Analysis*, Forthcoming Stanford CS Report, Nov.

Baumgart, Bruce G. [1974]: *Geometric Modeling for Computer Vision*, Stanford Artificial Intelligence Laboratory, Memo AIM-249, Oct.

Braid, I. C. [1973]: *Designing With Volumes*, Cantab Press, Cambridge, England.

Binford, Thomas O. [1971]: *Visual Perception by Computer*, Invited paper at IEEE Systems Science and Cybernetics Conference, Miami, Dec.

Brooks, Rodney A., Russell Greiner and Thomas O. Binford [1978]: *A Model Based Vision System*, Proceedings: ARPA Image Understanding Workshop, Cambridge, Mass, May, pp. 36-44.

Davis, Randall [1976]: *The Application of Meta-Level Knowledge to the Construction, Maintenance, and Use of Large Knowledge Bases*, Stanford Artificial Intelligence Laboratory, Memo AIM-283, Jul.

Davis, Randall, Bruce Buchanan and Edward Shortliffe [1975]: *Production Rules as a Representation for a Knowledge-Based Consultation Program*, Stanford Artificial Intelligence Laboratory, Memo AIM-266, Oct.

Grossman, David D. [1976]: *Procedural Representation of Three Dimensional Objects*, IBM Journal of Research and Development, Vol. 20, No. 6, Nov., pp. 582-589.

Miyamoto, Eiichi and Thomas O. Binford [1975]: *Display Generated by a Generalized Cone Representation*, Conference on Computer Graphics and Image Processing, May.

Nevatia, Ramakant and Ramesh Babu [1978]: Personal communication

Roberts, Bruce P. and Ira P. Goldstein [1977]: *The FRL Manual*, MIT Artificial Intelligence Laboratory, Memo 409, Sep.

Sutherland, Ivan E., Robert F. Sproull, and Robert A. Schumaker [1973]: *A Characterization of Ten Hidden-Surface Algorithms*, Evans and Sutherland Computer Corporation, Salt Lake City, Utah. (also published in ACM Computing Surveys, 6 (no. 1) March 1974)

Voelcker, H. B. [1974]: *An Introduction to PADL: Characteristics, Status, and Rationale*, University of Rochester, Production Automation Project Technical Memo 22, Dec.

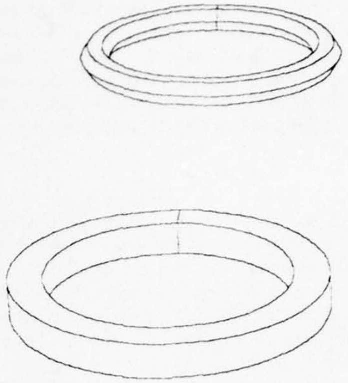


Figure 4.

```
Need to calculate: (RECTANGULAR WIDTH) for FUSELAGE
  Invoking rule: R19 in context FUSELAGE
    Need to prove: (RIBBON SHAPE) is RECTANGULAR for FUSELAGE
      Previously proven
    Need to prove: (CIRCULAR CROSS-SECTION) is T for FUSELAGE
      Invoking rule: R13 in context FUSELAGE
        Rule: R13 Failed

      Invoking rule: R12 in context FUSELAGE
        Concluding: (CIRCULAR CROSS-SECTION) is T for FUSELAGE
        Concluding: (CIRCULAR RADIUS) is 3.0 for FUSELAGE
        Rule: R12 Succeeded


    Need to calculate: (CIRCULAR RADIUS) for FUSELAGE
      Previously proven
    Concluding: (RECTANGULAR WIDTH) is 6.0 for FUSELAGE
    Rule: R19 Succeeded
```

Partial trace of the backward chaining mechaninsm.
Figure 6.

```
(MAK-RULE R19
    PREMISE (CAN-PROVE $CONTEXT (RIBBON SHAPE) 'RECTANGULAR)
            (CAN-PROVE $CONTEXT (CIRCULAR CROSS-SECTION) T)
            (CAN-CALCULATE $CONTEXT (CIRCULAR RADIUS))
    ACTIONS (CONCLUDE $CONTEXT (RECTANGULAR WIDTH)
                        (*$ 2.0(VALUE-OF $CONTEXT (CIRCULAR RADIUS))))
```

Definition of rule R19
Figure 7.

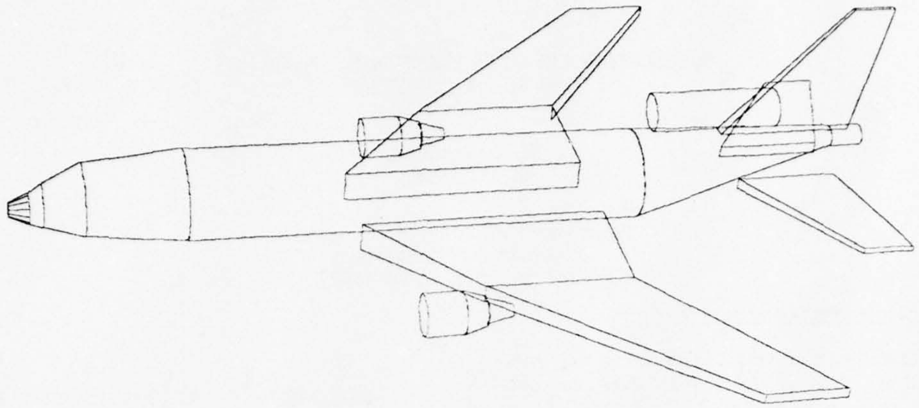
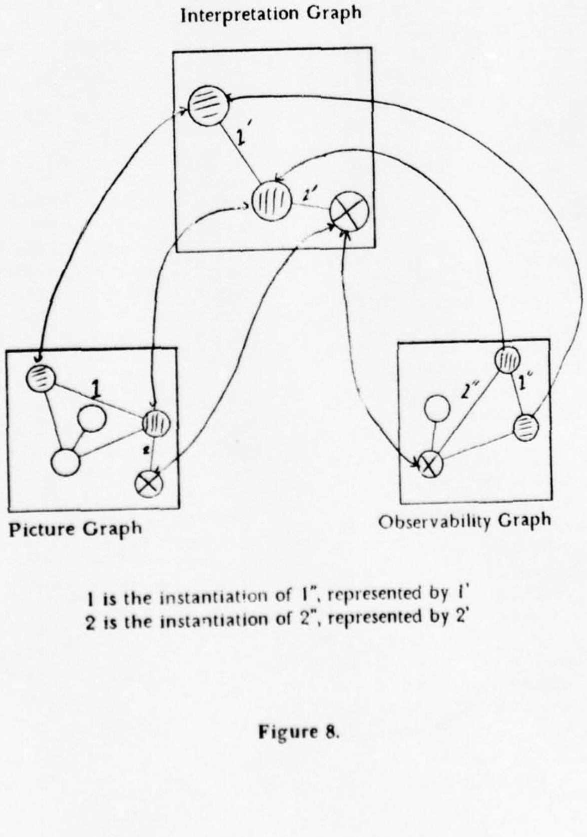


Figure 5.



1 is the instantiation of 1", represented by 1'
2 is the instantiation of 2", represented by 2'

Figure 8.

REPRESENTING AND USING LOCATIONAL CONSTRAINTS IN AERIAL IMAGERY*

D. M. Russell
C. M. Brown

Department of Computer Science
University of Rochester
Rochester, New York 14627

I. Constraint Networks

Constraint Networks are part of the high-level model in the Rochester Vision System [Ballard, et al.]. A Constraint Network (CN) models a real world object's expected location in an image by describing its relationships to other objects of known position. Each of these descriptions is a constraint on the object's location in the image. For instance, a dockyard is usually found adjacent to the water's edge and in or near a harbor. This statement tells us two characteristics of real dockyards: 1) dockyards are adjacent to the coastline; and 2) dockyards are in or near harbors. Both statements constrain the dockyard's possible location by specifying where it would be with respect to the coastline and to harbors. CN's are an embodiment of this kind of knowledge. In this report, CN's are used in the domain of image understanding to illustrate the more general principles involved in continual search-space refinement.

A CN is composed of nodes representing objects or object locations and arcs specifying operations which express constraints between them. Each constraint serves to determine the object location more precisely within the image by limiting the possible area where the feature could occur.

Specifically, a CN is the data structure which we use to represent these constraints. Normally, a CN serves as a data source giving the feature's location. However, since the facts which limit the possible location of the object are explicitly encoded by the nodes of the CN, if the location is not known when the CN is interrogated, then an evaluator can use the CN to compute the feature's location. When a CN is evaluated, the evaluator uses the knowledge encoded in the structure of the CN and data available to compute the most likely area in the image where a particular feature may be found. So, Constraint Networks offer an inexpensive way to eliminate large parts of the image from analysis by explicitly indicating where to look next when given some contextual clues. In many scenes, information about the location of one feature can specify the locations of others in the picture [Garvey].

However, when modeling real world constraints, we need not limit the kind of knowledge used to simple relations of feature locations within an image. CN's can also utilize additional information about the domain of interest. We might also know, for example, that docks have a normalized albedo above some determined value for aerial photographs. Knowing this fact would immediately reduce our searching for docks to those areas of the scene which have a reflectivity greater than the value specified. Domain-specific knowledge of this sort can also be represented as a constraint which limits the range of values assumed by a feature description.

While these kinds of knowledge can also be represented as a group of assertions, each encoding a single constraint, we choose the network format for representing our constraints because 1) it is a formal structure which can explicitly encode the relationships between features easily, 2) it provides a facility for optimization of evaluation and sharing of partial results, and 3) it is a simple way to compose complex constraints from primitive constraints in a straightforward manner.

*This paper is an abridgement of [Russell].

II. Constraint Networks: Structure and Function

The network is composed of nodes of three types:

Feature nodes - are the handle by which the Constraint Network is accessed by some larger image understanding system. The CN under a feature node embodies the knowledge which describes a particular feature in an image. A Feature Node has attached to it (as sons) CN's which are alternative encodings of the possible locations of the feature in the image. These CN's may be thought of as different strategies for finding the feature. Yet, a CN is not a completely procedural mechanism for representing knowledge, for associated with each node in a CN is the result of the evaluation of the CN below that node. This also holds true with the feature node; if the entire CN has been evaluated below the feature node, then the feature node contains the result of that evaluation. In this case, "evaluation" of the CN becomes a simple lookup. In other words, a CN is a "compute when required" structure which minimizes the amount of processing that it must perform. This is similar to the idea of "memo functions" as suggested by [Michie].

Operation nodes - are the nodes encoding the various constraints which are placed on the feature being searched for in the image. An operation node gets input from all of its sons and then applies the operation it represents on that data, thus realizing the constraint. Operation nodes represent the geometric relationships between features and are operations chosen from some system-defined set of primitives.

Data nodes - are the terminal nodes of the network. That is, they have no sons and always evaluate to data. Data nodes supply an unevaluated network with initial image data to operate on; they usually correspond to locations or image features which are relatively easy to determine.

The nodes of a CN can all potentially hold data. This capability is used to store the partial results found during an evaluation of the CN. As a result, all nodes are always in one of four states. A node is UP-TO-DATE if the data attached to it is a valid instance of the feature in the image. A node is OUT-OF-DATE if no data is attached to the node (i.e. it is not known if this primitive feature exists in the image); the node can be NONE-THERE if it is known that no primitive feature of this type exists in the image, or finally the node can contain information which is HYPOTHESIZED (the result of the evaluation of a CN and may not truly exist in the image). Each different status affects the results of node evaluation, and the way that results are handled by any nodes which use the result. A node that is OUT-OF-DATE returns a value which indicates that the answer may be anywhere in the UNIVERSE of the image. An UP-TO-DATE node explicitly points to the feature in the image. A node which is HYPOTHESIZED determines a location in the image, but the data may or may not locate the specified feature since it is the result of the evaluation of another CN. A HYPOTHESIZED result is considered the most likely location of a feature until the validity of the HYPOTHESIZED data can be verified. Finally, a node which is in a NONE-THERE state indicates that the feature simply doesn't exist in the image, or that all instances of that feature in the image have already been bound to other nodes which describe this feature. (This distinction is easy to make, but is only performed if required.)

III. Constraint Types

The operations encoding primitive geometric constraints are chosen from a set of basic operations which describe transformations on areas, describe relationships between areas, specify shapes and the like. The function of the operations in the primitive set is to provide the CN builder with enough tools to describe flexibly and naturally image areas and their relationships with other image areas. Although the number of potential operations is quite large, we have found that a small number of primitives (about twenty) suffice for most of our descriptive tasks.

In our system, the primitive set is made up of four different types of operations.

Directional operations specify where to focus attention. Operations such as LEFT, REFLECT, NORTH, UP and DOWN all constrain the sub-image to be in a particular orientation to another feature.

Area descriptions specify a particular area in the scene that restricts a feature location. For example, CLOSE-TO, IN-QUADRILATERAL, and IN-CIRCLE define areas at some location in an image of interest.

Set operations permit areas to be handled as point sets of pixels. These operations, such as UNION, DIFFERENCE and INTERSECTION make very complex image areas far easier to describe.

Predicates on areas allow features to be filtered out of consideration by measuring some characteristic of the data. For example, a predicate testing WIDTH, LENGTH or AREA against some value would restrict the size of features in consideration to be only those within a permissible range.

In actually constructing a CN, the builder is not limited to building CN's from purely primitive operations. Since a CN represents an implicit description of an area in the image, it can be used by other CN's as an operation for locating a feature. This allows the CN builder the ability to form very complex CN's by building upon previous work.

IV. Evaluation of Constraint Networks

A Constraint Network is evaluated by a special purpose evaluator working top-down in a recursive fashion, storing the partial results of each constraint at the topmost node associated with that constraint, with a few exceptions.

We can think of a CN's evaluation in the following way. Feature nodes may have several sons, or sub-CN's, each encoding a separate strategy. A particular strategy is selected by a strategist as described in [Lantz, et al]. The strategist computes the utility of each strategy attached to the feature node and based on this estimate the most desirable strategy is selected. The strategist's measurement of utility is based on both an a priori measurement of the algorithm's effectiveness and an assessment of the status of the data present in the body of the CN.

On this basis, the strategist interested in the feature node chooses a strategy and begins to evaluate the CN. Strategies of a feature node are evaluated until an answer is obtained or all strategies are exhausted.

When a strategy is selected, the root node of the strategy will be evaluated. In most CN's, this node will be an operation node. An operation node evaluates by first evaluating all of its arguments, and then applying its procedure to those results. Its own result is passed back to node of the CN which evaluated it. Of course, if the son of an operation node is a feature node or another operation node, then the evaluator will recursively continue to evaluate.

At some point in the course of the evaluation, the evaluator reaches a node which has already been evaluated and is marked UP-TO-DATE or HYPOTHESIZED (and therefore containing the results of evaluation below that point). The results of this node are returned and used exactly as if it were a data node. If a data node marked OUT-OF-DATE is evaluated, the evaluation mechanism returns a result stating that the primitive feature location is not specified, and that more work needs to be done by an executive procedure (which will presumably direct a low-level worker to find the needed information). If the primitive feature is then not supplied, the strategist will specify the status of the node. In this case, either the feature doesn't exist, all instances of that feature in the image are already bound to other nodes, or the worker could have simply used up all available resources before being able to return an answer. In the first two cases, the node would then be marked as NONE-THERE, and would return NIL to indicate that the feature desired is not in this image. Alternatively, if the worker has exhausted its resources, but has not yet determined the status of the feature described by the node, the node will remain OUT-OF-DATE and have the entire image as its value. This indicates that the feature could be located anywhere in the image. At a later date, processing can resume from this node without having to recompute the part of the tree which was already processed. Finally, a node marked HYPOTHESIZED has data which was inferred by a CN somewhere down the line of inference. HYPOTHESIZED data can and is used to make inferences, but the results of all inferences based on hypothesized data are marked HYPOTHESIZED as well.

V. An Example of CN Evaluation

Figure 1 - a CN for new construction site in an oil field

Figure 1 is the graphic representation of a Constraint Network which computes the area of probable new construction of tanks at a tank farm site. This CN embodies the strategy that of "New construction of oil tanks is found near the old oil tanks, but not very close and not on the old tanks themselves".

Set operations permit areas to be handled as point sets of pixels. These operations, such as UNION, DIFFERENCE and INTERSECTION make very complex image areas far easier to describe.

Predicates on areas allow features to be filtered out of consideration by measuring some characteristic of the data. For example, a predicate testing WIDTH, LENGTH or AREA against some value would restrict the size of features in consideration to be only those within a permissible range.

In actually constructing a CN, the builder is not limited to building CN's from purely primitive operations. Since a CN represents an implicit description of an area in the image, it can be used by other CN's as an operation for locating a feature. This allows the CN builder the ability to form very complex CN's by building upon previous work.

IV. Evaluation of Constraint Networks

A Constraint Network is evaluated by a special purpose evaluator working top-down in a recursive fashion, storing the partial results of each constraint at the topmost node associated with that constraint, with a few exceptions.

We can think of a CN's evaluation in the following way. Feature nodes may have several sons, or sub-CN's, each encoding a separate strategy. A particular strategy is selected by a *strategist* as described in [Lantz, et al]. The strategist computes the utility of each strategy attached to the feature node and based on this estimate the most desirable strategy is selected. The strategist's measurement of utility is based on both an *a priori* measurement of the algorithm's effectiveness and an assessment of the status of the data present in the body of the CN.

On this basis, the strategist interested in the feature node chooses a strategy and begins to evaluate the CN. Strategies of a feature node are evaluated until an answer is obtained or all strategies are exhausted.

When a strategy is selected, the root node of the strategy will be evaluated. In most CN's, this node will be an operation node. An operation node evaluates by first evaluating all of its arguments, and then applying its procedure to those results. Its own result is passed back to node of the CN which evaluated it. Of course, if the son of an operation node is a feature node or another operation node, then the evaluator will recursively continue to evaluate.

At some point in the course of the evaluation, the evaluator reaches a node which has already been evaluated and is marked UP-TO-DATE or HYPOTHESIZED (and therefore containing the results of evaluation below that point). The results of this node are returned and used exactly as if it were a data node. If a data node marked OUT-OF-DATE is evaluated, the evaluation mechanism returns a result stating that the primitive feature location is not specified, and that more work needs to be done by an executive procedure (which will presumably direct a low-level worker to find the needed information). If the primitive feature is then not supplied, the strategist will specify the status of the node. In this case, either the feature doesn't exist, all instances of that feature in the image are already bound to other nodes, or the worker could have simply used up all available resources before being able to return an answer. In the first two cases, the node would then be marked as NONE-THERE, and would return NIL to indicate that the feature desired is not in this image. Alternatively, if the worker has exhausted its resources, but has not yet determined the status of the feature described by the node, the node will remain OUT-OF-DATE and have the entire image as its value. This indicates that the feature could be located anywhere in the image. At a later date, processing can resume from this node without having to recompute the part of the tree which was already processed. Finally, a node marked HYPOTHESIZED has data which was inferred by a CN somewhere down the line of inference. HYPOTHESIZED data can and is used to make inferences, but the results of all inferences based on hypothesized data are marked HYPOTHESIZED as well.

V. An Example of CN Evaluation

Figure 1 - a CN for new construction site in an oil field

Figure 1 is the graphic representation of a Constraint Network which computes the area of probable new construction of tanks at a tank farm site. This CN embodies the strategy that of "New construction of oil tanks is found near the old oil tanks, but not very close and not on the old tanks themselves".

corresponding to the desired feature depends to a large part on the number of UP-TO-DATE nodes which it uses during its evaluation.

VII. Grain Size of Constraint Networks

In the hierarchy of data structures produced during image analysis, the level of effective operation of Constraint Networks seems largely limited by the nature of the expectations incorporated into the CN. The constraints used are static by nature and their applicability seems limited to high-level concepts. We have found that CN's tend to become difficult to manage effectively at low levels of domain representation and inferencing. In the vision domains we have studied, low level processing such as region growing, edge following or the like do not seem easily amenable to representation as Constraint Networks. CN's can easily provide an adequate mechanism for line following when the edges are of high contrast. But in noisy environments and at small grain size, the strong interconnections between features rapidly become weak, reducing the basis on which Constraint Networks operate.

VIII. Future Work

A desirable future extension of Constraint Networks would be to incorporate some notion of the connection between structure and function in computing an object's most likely location in a scene. This would initially require that the CN perform inferencing of a different sort about the structure of a feature. Currently, the knowledge in a CN is structurally oriented. It describes the location of an object based solely on its relationships to other objects in 3-space. Comprehension of the functional connections between objects would greatly increase the robustness of feature location.

The dynamic data attachment mechanism is fairly expensive if the semantic description part fails, since it involves sophisticated graph matching between the input description and portions of the CN. This could be a substantial area for improvement.

Constraint Networks can also be used as a knowledge source describing the relationships between objects in an image. In this use of CN's, they act as a static representation of the interconnections between items, separating features from their functions in CN's. We have made some preliminary efforts in this direction, attempting to categorize the nature and manner in which non-geometric inferences could be made from the structure and contents of the CN.

IX. An Application

Figure 2 shows a CN incorporating two constraints on the location of aeration tanks in a water treatment facility:

Constraint 1: "Aeration tanks are located somewhere close to both the sludge tanks and the sedimentation tanks."

Constraint 2: "Aeration tanks must not be too close to either the sludge or sedimentation tanks."

Figure 2

In the case that we are able to start the CN with only a single sludge tank and a single sedimentation tank, the result of evaluation is shown in Figure 3.

Figure 3

A feature node is represented as a simple box; an operation node as a simple box divided by a horizontal midline; and a data node as a box within a box.

If we now add to the CN the location of the remaining sludge and sediment tanks in the picture, and re-evaluate the network, the result more accurately reflects the actual location of the aeration tanks. (Figure 4).

Figure 4

X. System Implementation

The CN system reported here was written in SAIL at the University of Rochester by Dan Russell during the summer of 1978. It consists of three programs - CNGEN, the Constraint Network Generator; PIC, the data set constructor and EVAL, the CN evaluator and test program. The programs communicate via disk files containing the LEAP world which defines not only the CN's, but the data as well.

Data is represented in LEAP as the datum of an item. Features in the picture are represented by lists of the pixel locations which the feature occupies. The canonical representation used is basically a run-length encoding of horizontal scan-line segments making up the region. This representation has several nice properties from an implementation viewpoint. It is very easy to represent multiple areas, or a discontinuous feature in a scene in a single list datum. Union, difference and intersection of areas are all straight-forward to implement, and the merge-like algorithms used run in time varying linearly with the size of the regions. Facts about the data contained in a data node are encoded as LEAP triples (or associations) which state a particular quality of the data node. The triples assert facts such as data type (the representation used; INTEGER, AREA, REAL) node name, node status (HYPOTHESIZED, OUT-OF-DATE, NONE-FOUND, UP-TO-DATE), and which nodes are sons or fathers of a given node.

Constraint Networks are also represented in LEAP. In the same way, LEAP triples are used to represent the Father-Son relationships between nodes in a network and to associate the various node states with each node. In LEAP notation, a node which was out of date would be -

```
VALIDITY of NODE is OUT!OF!DATE
```

The process of generating the CN's and saving their structure onto disk is done by the CNGEN program. This program runs interactively on a Grinnell color display, allowing the CN builder to see the CN's as they are being made. The program permits the builder to edit, create and delete CN's easily and quickly. The desirability of such a facility for semantic networks was recognized in [Brachman].

PIC takes digitized images and creates the initial data nodes for the CN's to evaluate. PIC can create arbitrary shapes interactively by using various sizes of circles, arbitrary quadrilaterals and lines. Complex shapes are formed by merging together smaller pieces of the shape to form the final region.

Finally, EVAL performs the evaluation of the CN's. EVAL accepts data sets and CN's on demand. It offers tracing facilities which display the result of the evaluation of each node in a different color. This facility makes it easy to follow the inferencing patterns of the CN in use and permits an easy way to follow the actions of the strategist.

This research was partially supported under DARPA grant N00014-78-C-0164.

References

[Ballard, et al] Ballard, D.H., C.M. Brown, and J.A. Feldman. "An Approach to Knowledge-Directed Image Analysis", TR21, Computer Science Department, University of Rochester, September 1977; also in *Proc. 5th IJCAI*, MIT, August 1977

[Barrow,et al] Barrow, H.B., Amber, A. P., Burstall, R. "Some Techniques for Recognizing Structure in Pictures", *Frontiers of Pattern Recognition* (ed. Watanabe, S.), pg. 1-19, Academic Press, New York

[Garvey] Garvey, T. D. "Perceptual Strategies for Purposive Vision", SRI Artificial Intelligence Center Technical Note 117, September 1976

[Lantz,et al] Lantz, K. A., C.M. Brown, D.H. Ballard, "Model-Driven Vision Using Procedure Description: Motivation and Application to PhotoInterpretation and Medical Diagnosis", *SPIE proceedings August 1978*

[Michie] Michie, D. "Memo Functions and Machine Learning", *Nature*, vol. 218, pg. 19-22, 1968

[Russell] Russell, D. R., "Constraint Networks: Modelling and Inferring Object Locations from Constraints", TR 38, September 1978.

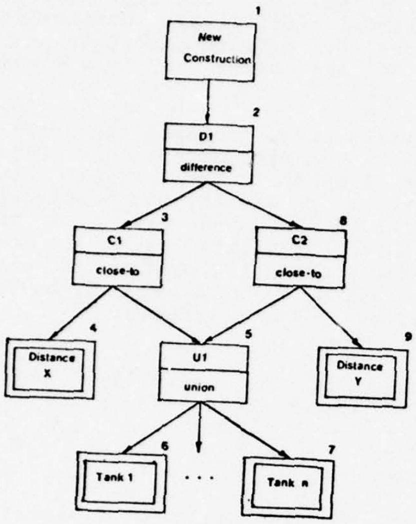


Figure 1

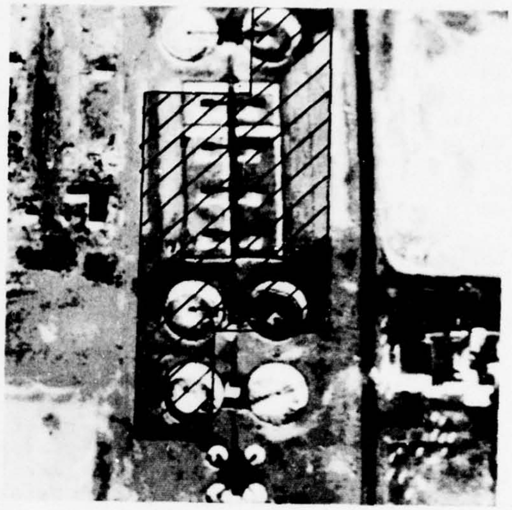


Figure 3

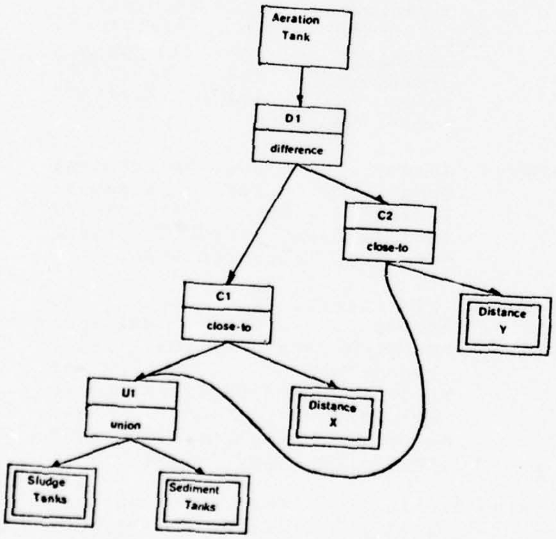


Figure 2

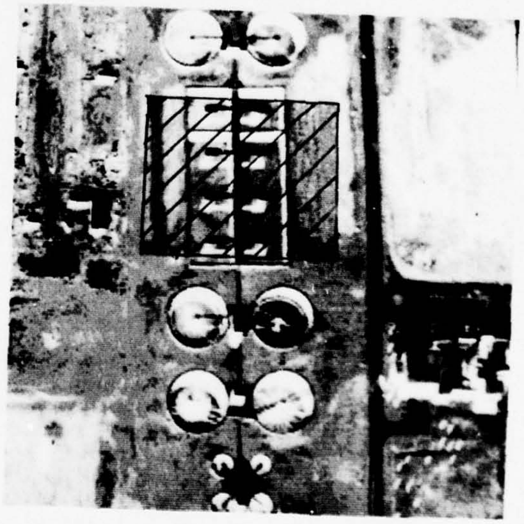


Figure 4

THE ARGOS IMAGE UNDERSTANDING SYSTEM

Steven M. Rubin

Department of Computer Science
Carnegie-Mellon University, Pittsburgh, Pa. 15213

Abstract

This paper reviews and presents the current results of the ARGOS Image Understanding System ARGOS demonstrates the feasibility of using a best-few, non-backtracking beam search technique on a uniform representation of knowledge. The system has recently been modified to use automatic segmentation and hierarchically organized knowledge. In addition, ARGOS has been successfully used to determine the angle of view of photographs taken from around the city of Pittsburgh.

Introduction

ARGOS is a computer system which can employ large and diverse amounts of knowledge to interpret images. Therefore it is an image understanding system. This paper briefly reviews the development of ARGOS and presents the most recent results that have been obtained. For a complete explanation of the workings of ARGOS, see the author's thesis [Rubin, 1978].

The basic system consists of three main sections: the knowledge, the image, and the search. The knowledge section builds a network structure which contains both generalizations and specific instances of information obtained from the knowledge sources. ARGOS currently uses color, texture, adjacency, occlusion, location, size, and a number of shape factors. The image section processes incoming photographs so that they can be matched to the network structure produced by the knowledge section. The final section is the search which uses a best-few, non-backtracking technique called Locus to match the image to the knowledge network. This match is the heart of ARGOS.

The current implementation of Locus search is conceptually similar to Markov processes. The premise is that an area of the image can be evaluated solely in the context of its immediate neighbors. If this inductive assumption is properly implemented, then a single scan of an image will produce an evaluation which takes the entire image into account. In addition to the formal property of an inductive chain, Locus includes a number of heuristics which reduce the search space without damaging the results. And, of course, the ARGOS implementation of Locus contains special modifications which enable the search to work in a two-dimensional domain.

Before proceeding any further, it is useful to explain the general approach that ARGOS takes towards image understanding. The basic problem is that knowledge about a three-dimensional scene must be matched to a two-dimensional image to produce an interpretation. Kanade [1978] shows two fundamental ways that this can be done. The first is to extract a three-dimensional structure from the image which is matched to the knowledge to form an interpretation. The alternate technique, which is used by ARGOS, is to generate two-dimensional projections of the knowledge and match them to the image. Both of these techniques match a form of the image to a form of the knowledge, and both involve conversion between a three-dimensional scene and a two-dimensional image of that scene. The former technique converts from 2-D to 3-D and matches in 3-D. The latter technique converts from 3-D to 2-D and matches in 2-D. This distinction is useful in understanding the design decisions of ARGOS.

One interesting design decision of ARGOS is its use of hierarchies of knowledge. Since image understanding is very complex, it is impossible for one search tree to employ all of the knowledge in a scene. Therefore the knowledge is hierarchially divided from the general to the specific. Each pass of Locus search applies general knowledge and uses the results to select a less general knowledge network for the next pass.

ARGOS is currently working on two levels of the knowledge hierarchy for the task of interpreting photographs of downtown Pittsburgh. The top level is the more general task of identifying the angle of view around the city from which the photograph was taken. In over a dozen photographs, the system pinpointed the view with an average error of 41 degrees. The bottom level of hierarchy uses the selected view to generate more specific knowledge about the photograph.

The rest of this paper discusses the knowledge used by ARGOS, the search process, hierarchies of knowledge networks, and the current results.

Knowledge

All of the knowledge used by ARGOS is placed in a network. The nodes of the network represent areas of an image and the arcs connecting the nodes represent relationships between the areas. At a gross level, then, knowledge can be divided into two classes: that which belongs in the nodes and that which belongs on the arcs.

Most knowledge appears in the nodes of a knowledge network. For example, the *color* and *texture* of a building is stored in the node for that building. ARGOS actually implements this as a varying number of color/texture templates for each building. The system tries to keep enough templates to cover all of the appearances of the building so that precise identification can be made.

Another knowledge source that is stored in network nodes is shape. ARGOS uses four shape operators to describe a region [Price, 1976]. The *fractional fill* is the ratio of the region area to the size of its minimum bounding rectangle; thus it is a measure of how tightly packed the region is. *Compactness* is the ratio of the perimeter squared to the area of a region. It describes the nature of the region edge and also indicates how irregularly shaped the object is. *Orientation* is the angle of an elongated region, and *elongation* specifies the ratio of length to width. Both orientation and elongation are derived from the first moment of the Fourier transform of the region.

There are many other types of knowledge which are lodged in the network node. Object *location* within the image is one. Absolute and relative object *size* is another. All of these knowledge sources have the property that they describe a particular object in the image and they describe it independent of image context. The contextual information is encoded in the arcs which connect the nodes.

In a purely two-dimensional sense, network arcs embody only one knowledge source: *adjacency*. The presence of an arc indicates an adjacency between the nodes that it connects. In addition, there is information on the arc which specifies the nature of the adjacency. The only information that ARGOS uses is the direction of adjacency, but it is possible to imagine many other modifiers such as edge texture, edge angle, and relative proximity.

A comparison between the wealth of knowledge in the nodes and the dirth of knowledge in the arcs might lead to the conclusion that networks are unnecessary and that a simple template match can do as well as ARGOS. This is not the case for two reasons. One reason, which will be discussed later, is that the search process uses the network arcs to guide and speed the interpretation.

Another reason that networks are important is that the adjacency information embodies many other knowledge sources. Recall that knowledge starts as three-dimensional data and is projected to a two-dimensional image for matching purposes. When this projection is done, much of the spatial information is preserved in the adjacency. Therefore network arcs actually contain knowledge about structure, occlusion, shadows, and other spatial knowledge.

In general, it is easy to find knowledge and incorporate it into the network nodes, but it is harder to find knowledge for the network arcs. This is due to the difficulty of converting from a three-dimensional model to a two-dimensional view without losing most of the three-dimensional knowledge. ARGOS typically builds a number of views of the model. Each view is a separate network whose arcs define the relationship of objects in that view. ARGOS then combines all of the view networks into one large network which has general and specific knowledge about all of the views.

A number of simple rules exist for the generalization of information in the view networks so that the final knowledge network can be built. For example assume that two views of the city show the Hilton Hotel, and that the surrounding context of the Hilton is identical in both views. The generalization rules will merge the two Hilton nodes and combine all of their arcs in the knowledge network. This reduction in the number of nodes is desirable because it makes networks smaller and faster to search. Also, by combining multiple views into one network, a single network path can include options from many classes of views, thus allowing general knowledge to be applied.

When generalizing two network nodes from different views, ARGOS requires that at least 70% of their adjacencies match before they will be merged. In addition, the size and location are taken into account so that radically different views with similar context will remain separate in the knowledge network. The generalization process typically reduces the number of network nodes by 60%. The completed network contains all of the knowledge about the scene that can be gleaned from the projected views.

Search

The basic premise underlying Locus is that the problem of image interpretation can be viewed as a problem of search. Given a knowledge base in network form and an unknown image, Locus finds the path through the knowledge which corresponds to the image. This path defines a labeling for the image network and is therefore an interpretation of the image.

Locus proceeds by building a highly pruned search tree of alternative paths through the knowledge network. Each level of depth in the search tree corresponds to one of the nodes in the image network. An image which is divided into 50 segments will cause Locus to generate a search tree that is 50 levels deep. At each level, there are a number of alternatives which are taken from the knowledge network. Locus must select exactly one alternative at each level to find the correct knowledge network path.

Finding the correct set of knowledge network nodes is a combinatoric problem without Locus search. Locus currently uses the Markov assumption which allows it to evaluate the choices for each depth level solely in terms of the previous depth levels that physically adjoin the current one in the image network. When the entire search tree has been evaluated in this manner, a re-examination of the tree quickly finds the optimal set of labels.

Although the one-pass search is a major factor in speeding the interpretation process, it could not gain this advantage without the knowledge network. Each step of Locus search begins by examining the neighboring depth levels in the search tree. The knowledge network arcs from these neighbors are used to select an initial set of nodes that may exist at the current depth level. This list is then evaluated and pruned. The important thing to notice is that the knowledge determines the complexity of the search because the knowledge network arcs select the labeling candidates. Thus the network concept is very important to Locus.

Hierarchies

To summarize, ARGOS is able to hypothesize a set of views of a scene and combine the knowledge from these views into a network. It can then match this network to an image. The result is a labeling which is derived from one or more of the initial views of the scene.

There are two ways of considering this process. The simplistic approach is that given enough hypothesized views, a network can be built with all of the knowledge about a scene. The more realistic approach is to treat the process as one of *givens* and *unknowns*. Each run of Locus search starts with all given knowledge and hypothesizes enough views to cover the unknowns that it wishes to determine. The results of this run help to resolve the unknown allowing a new knowledge network to be built. By iterating Locus search, ARGOS is able to step through a series of questions, answering them one at a time to form a complete understanding of the image.

For example the current ARGOS task has the givens that the image is Pittsburgh during the day in the winter, but it does not know the angle or distance of view, the names of each object in the image, etc. The first question that is asked is the angle of view around the city. To answer this question, ARGOS builds a network which contains multiple views of the city from different angles. The results of Locus search on this network will identify the view angle by specifying a likely path through the network. This information is then used to build a more detailed network so that the next question can be answered: "what are the names of the objects in the photograph?" For this question, the view angle is no longer an unknown; it is given.

The process of identifying unknowns, selecting from them, and then moving on to the next unknowns is called knowledge hierarchy traversal. ARGOS views this hierarchy as being organized from the general to the specific. The top of the hierarchy is very general and is the question that is asked first. In the two-level hierarchy mentioned above, the top level is the view angle identification task. It is more general and it must be answered first. When the angle of view is known, then the low level of the hierarchy is run. It is more specific and answers the question of object identification. The only issue that is not well understood is how knowledge from one level of the hierarchy is transmitted to the lower level.

There are a number of ways of transmitting hierarchical network knowledge. The obvious way is to use the results of upper level networks as a knowledge source for the lower level network. If the view angle task determines that the photograph was taken from the west of the city, then the next pass of Locus can use the same network, but penalize any transitions to nodes that aren't part of the western view. The use of hierarchy results as a network transition penalty has the advantage of fitting in well with the general use of knowledge in Locus since all other knowledge sources are implemented as transition penalties. This technique also has the advantage that the lower level network need not be built from the results of the upper level search, but can be statically computed. The drawback to the use of upper level results as a knowledge source is that the lower level network becomes unnecessarily large: it must contain two levels of knowledge, one of which is mostly ignored because it repeats the upper level network. Another problem with this scheme is that the excess information in the lower level network is confusing to the search, even though it is guided by a knowledge source.

The alternative way of applying hierarchical network knowledge is to re-build the network. This allows more new knowledge to be employed since there is less knowledge carried over from the upper level network. The problem with this technique is that the system must make the correct choice at each level of the hierarchy or else the lower level will be stuck with a very detailed network that is totally incorrect. This is a standard tree search problem.

So far, there are no good solutions to the problem of knowledge hierarchy traversal. However there is no reason to doubt that it can be made to work.

Current Research And Results

Investigations of ARGOS are focusing on two points of interest. The first is the use of automatic segmentation to speed the processing and improve labeling accuracy. The second area of research is the use of hierarchies of knowledge networks. Since these are ongoing explorations, the current results will only partially reflect the virtues of these features.

Before automatic segmentation, the input images were 75 by 100 pixels in size. Search trees for these 75 by 100 images were, of course, 7500 deep. This meant that the search process took a long time and that the inductive search assumption was numerically unstable.

ARGOS has recently been modified to accept arbitrarily shaped segments. The experiments presented here are with hand-drawn segments, but the system will soon be running with images that are automatically segmented with a clustering algorithm [Shafer and Kanade, 1978; Ohlander, 1975]. Typically, images are broken down to 50 segments. This makes the search much faster and allows tighter constraint of knowledge.

The other current investigation is the hierarchical use of knowledge. In order to explore knowledge hierarchies, it was necessary to formulate the view angle task for ARGOS. This task involved building a network of 24 views of the city from constant distance and elevation, with 15 degrees of lateral angle between each view, thus spanning a full 360 degrees around the city. Although the internal model of the city was composed of the 58 objects listed in Table 1, it was found that this much detail was useless in the view angle identification task. This is because view angle is determined from gross characteristics such as skyline and the relative positions of significant buildings and rivers. Therefore, it was necessary to generalize the knowledge and reduce the number of labels. The starred objects in Table 1 were selected for the view angle identification task. These labels were generated automatically in a process that examined all of the machine-generated views of the city and determined the significant buildings from the number of times that each one appeared in the skyline.

Table 2 shows the results of the view angle identification task on fifteen photographs of the city. Seven of these photographs were used to tune the parameters of ARGOS and the remaining eight were saved for test purposes. Naturally the training photographs scored better with an average error of only 30 degrees in the view angle identification. The test images were off by an average of 51 degrees. All of these photographs are reproduced in color in Appendix I of the author's thesis.

Conclusions

ARGOS is an interesting exploration of knowledge representation and search. It can accurately identify the view angle of photographs of the city of Pittsburgh and it can identify the objects in the photographs. Future investigations will concentrate on the use of more knowledge at all stages. In addition, ARGOS is being re-coded so that it will run on a PDP-11 system with writable microstore running the UNIX timesharing system. Although it currently requires a minute of CPU time on a PDP-KL10 computer, some improvement may be obtained on this new system. Regardless of hardware, ARGOS has demonstrated the merits of Locus search and a uniform representation of knowledge in image analysis.

References

Kanade, T., 1978. Task Independent Aspects of Image Understanding, Proceedings of the May, 1978 Image Understanding Workshop, Lee Baumann, ed, 45-50.

Ohlander, R., 1975. Analysis of Natural Scenes, (Ph.D. Thesis, Carnegie-Mellon University), Tech. Report, Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pa.

Price, K., 1976. Change Detection and Analysis of Multispectral Images, (Ph.D. Thesis, Carnegie-Mellon University), Tech. Report, Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pa.

Rubin, S., 1978. The ARGOS Image Understanding System, (Ph.D. Thesis, Carnegie-Mellon University), Tech. Report, Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pa.

Shafer, S. and Kanade, T., 1978. KIWI: A Flexible System for Region Segmentation, Tech. Report, Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pa. (in preperation).

Table 1: Labels

These 58 objects are the basic units of labeling for the low level of the knowledge network hierarchy: the object identification task. The starred objects are used in the high level of the hierarchy: the view angle identification task.

Alcoa Bldg
*Alleghony River
Alleghony Towers Bldg
Bell Telephone Co Bldg
Blue Cross Bldg
Eighth Ave Parking
Equibank Bldg
Federal Bldg
Fort Duquesne Blvd
Fort Duquesne Bridge
Fort Pitt Blvd
Fort Pitt Bridge
Fulton Bldg
Gateway Center Bldg 1
Gateway Center Bldg 2
Gateway Center Bldg 3
Gateway Center Bldg 4
Gateway Towers Apts
Gimbels Dept Store
*Grant Bldg
*Gulf Bldg
I. B. M. Bldg
Jenkins Arcade Bldg
Joseph Hornes Dept Store
Koppers Bldg
*Mellon Natl Bank Bldg
*Miscellaneous Buildings
*Miscellaneous Bridges
*Miscellaneous Roads
*Monongahela River
Mountains
Ninth Ave Bridge
Ninth Ave Parking Garage
*Ohio River
Oliver Bldg
*One Oliver Plaza
*Park
Penn Technical Center
Pennsylvania State Office Bldg
Pennsylvania State Office Lobby
Penthouse Apartments
Pick Roosevelt Hotel
Pittsburgh Hilton Hotel
Pittsburgh Natl Bank Bldg
Pittsburgh Natl Bank Operations
Pittsburgh Press
Rust Bldg
Shields Rubber Bldg
Sixth Ave Bridge
Sixth Ave Parking Garage
*Sky
Snow
Stanwix St Bridge Remnants
*Three Rivers Stadium
*U. S. Steel Bldg
United Engineering Bldg
*Westinghouse Bldg
Westinghouse Plaza

Table 2: View Angle Identification Results

This table shows the results of the fifteen images that were used in the view angle identification task of ARGOS. The training images were used to tune the system; the test images were run once the system was tuneo. Note that all angles are expressed in 15 degree increments since that is the granularity of the task.

Image	True Angle	ARGOS Guess	Error
Training 1	300-315	300	0
Training 2	300	330-345	30
Training 3	240-255	330-345	75
Training 4	0-15	15	0
Training 5	0-15	330	30
Training 6	345	0	15
Training 7	45-60	330-345	60
Training Average			30
Test 1	315	195	120
Test 2	285-300	330-345	30
Test 3	255	240	15
Test 4	300-315	240	60
Test 5	45	0	45
Test 6	45-60	135	75
Test 7	45-60	0	45
Test 8	15-30	0	15
Test Average			51
Overall Average			41

THE SRI ROAD EXPERT:
IMAGE-TO-DATABASE CORRESPONDENCE

R.C. Bolles, L.H. Quam,
M.A. Fischler, H.C. Wolf
SRI International
Menlo Park, California

ABSTRACT

Given an image to be analyzed and an approximate correspondence between the image and a map database, one of the important subtasks required of the road expert is to improve the correspondence. The basic refinement uses the database to predict the locations of known features, uses detection techniques to locate these features, and uses the feature matches to refine the correspondence. In this paper we describe new techniques for some of the important computations in this process. In particular, we discuss a technique to predict a region in the image within which a feature is expected to appear, a set of techniques to verify feature matches, and a technique to extend the refinement process to include a new type of match based on linear features, such as roads, which are prominent in the road domain. These techniques are demonstrated in an example in which the system reduces the uncertainties from approximately plus or minus 200 feet on the ground to approximately plus or minus two feet.

INTRODUCTION

Computing an image-to-database correspondence is a general problem occurring in all knowledge-based systems. In most image tasks the correspondence is a projective transformation and can be modeled as a function of the camera parameters, such as focal length, X, Y, Z, heading, pitch, and roll. If the parameters are known precisely, the model can precisely predict the two-dimensional image coordinates for any three-dimensional database point.

One common form of the image-to-database correspondence problem is to be given good estimates of the camera parameters and be asked to improve them. This task is important in many military situations. For example, in navigation it is the crucial step that improves the system's estimate of the location of the plane or missile. In change detection it is used to align two images of the same area so that the corresponding regions can be compared. In the Road Expert (see the companion paper by Fischler for an overview of the SRI Road Expert [8]) it is the key to the utilization of the database in subsequent tasks such as road monitoring.

The basic approach we are using to refine a correspondence is to locate known features in the image and use their locations to improve the correspondence (see Figure 1). The database contains descriptions of the available features. From these descriptions a set of features is chosen to be located that is based on the predicted viewpoint and viewing conditions. The estimates of the camera parameters are used to predict what the features look like and where they are likely to appear. Feature detection techniques ("operators") are chosen to locate the features and they are applied. Since the operators may not locate their intended features, their results are verified either by locating a larger portion of the features or by checking the relative positions of other features. After a set of features has been found, their locations are used to refine the estimates of the camera parameters. The parameters are refined by searching the parameter space for sets of parameter values that minimize the distances between the predicted locations of features and the locations determined by the operators. If the correspondence is not precise enough, the whole process can be repeated.

The important computations and decisions required to refine a correspondence are listed below:

(1) selection of features

(2) prediction of the appearance of a feature

(3) selection of an operator to locate the feature

(4) prediction of the nominal image location of a feature

(5) prediction of the range of image locations about a feature's nominal location

(6) selection of the order in which to apply the operators

(7) application of the operators

(8) verification of the results produced by an operator

(9) decision of when to use the results of one or more operators to help other operators locate their features

(10) decision of when to update the whole correspondence

(11) computation of a refined correspondence

(12) decision to stop

A number of people have worked on individual items in this list [1, 3, 4, 5, 6, 7, 9, 10, 11, 13], but mainly for pairs of images that were taken closely in time and from similar viewpoints.

There are several factors in the military domain, as well as other domains, that increase the difficulty of these items beyond current capabilities. Examples of such factors are a wide variety of viewpoints, a distribution of shadows, and the possibility of clouds. All of them make it more difficult to select features, predict the appearance of features, and locate features. Therefore, they increase the need for feature verification and strategy decisions. Which operators should be used for an image taken from this viewpoint and under these conditions? When should the results of one operator be used to reduce the predicted search area for a nearby feature? This type of question becomes more important as features become harder to find.

Our research goal is to produce an automatic system to refine correspondences within the road domain. To reach this goal we need to develop new models and techniques for several of the items in the above list. So far we have concentrated on a few of them: the prediction of the range of image locations for a feature, the verification of the results of an operator, and the computation of a refined correspondence. In this paper we will state our assumptions, describe our new techniques, and present an example.

ASSUMPTIONS

Our assumptions are summarized in Figure 2.

Figure 3 is a typical picture to be processed by the system. We assume that the resolution of the digital images will be between 20 feet/pixel and 1 foot/pixel. Figure 4, which is another picture of the site shown in Figure 3, is displayed so that one pixel corresponds to approximately sixteen feet on the ground. Figure 5 is a portion of Figure 3 displayed at its full resolution of approximately 1 foot/pixel.

We assume that we will have a database of the area on the ground contained in each picture to be analyzed. The database contains the geometry and topology of the roads and the locations of other features, such as road markings. Since we expect to obtain repetitive coverage of the areas of interest, the database may also contain information about the appearances of the road sections and features derived from previous images.

Images of the same site may be taken at different times of the day so the shadows may be different. Notice the variation in shadows between Figures 3 and 4. Part of the information expected by the system for each picture is the day of the year and the time of day at which the picture was taken.

Some of the images may contain clouds that obscure some of the roads and other database features (e.g., see Figure 6). And more generally, terrain features, buildings, and trees may obscure features of interest. The implication is that the system should be able to handle operators that find multiple matches, incorrect matches, or no matches at all.

Different pictures of the same region may be from different viewpoints. In particular, they may be from significantly different altitudes (e.g., twice as high) or different angles (e.g., 45-degree obliques versus vertical pictures). Figures 3 and 4 are pictures of the same site except that Figure 4 was taken from approximately twice the height and at a heading that is different from that of Figure 3 by almost 90 degrees. The wide variety of viewpoints implies that intensity correlation is not always sufficient to locate features. Other operators will be necessary.

Even though the viewpoint may vary widely, we expect to be given good estimates of the camera parameters for each picture. The camera parameters can be factored into two convenient sets: internal camera parameters and external camera parameters. The internal parameters describe the camera-specific information, such as the focal length of the lens. The external parameters describe the relative position and orientation of the camera with respect to the world represented in the database. Generally, the a priori estimates of the internal parameters are much better than the estimates of the external parameters.

We expect a measure of the uncertainty associated with each parameter estimate. For example, the HEADING might be estimated to be 75 degrees, plus or minus one degree. These uncertainties are used to predict the regions in a picture to be searched in order to locate a feature. We will refer to these search regions as "uncertainty regions." The smaller the uncertainties, the smaller the uncertainty regions; the smaller the uncertainty regions, the easier it is to automatically locate the desired features.

Two of our most important assumptions restrict the range of initial uncertainties about the camera parameter estimates. The first one restricts the combined internal and external uncertainties so that they do not imply uncertainty regions on the ground of more than approximately plus or minus 200 feet. The second one restricts the size of each parameter's uncertainty so that it is relatively small. The first assumption, in effect, restricts the sizes of the uncertainty regions that have to be searched to locate a feature. For example, if an image has a resolution of 1 foot/pixel, the largest uncertainty region would then be approximately 400 x 400 pixels. The second assumption limits the portion of the parameter space that the optimizer has to search. It also indirectly limits the maximum geometric change in the appearance of a feature.

An implicit assumption behind the characterization of a correspondence as a function of the camera parameters is that the imaging process can be modeled as a perspective transformation. If it cannot, a different mapping function would have to be used, but the same numerical approach would apply.

UNCERTAINTY REGIONS

Given parameter estimates and uncertainties about those estimates, where in the image is a feature likely to appear? Or more specifically, what region in the picture will have a given probability (e.g., a 95% probability) of containing the feature? To answer this question, one has to predict the effect on the location in the image of a feature caused by changing the parameter values in accordance with their stated uncertainties. To do that, one needs a model of their uncertainties. The error model we use is that the parameters vary according to a joint normal distribution, which is a reasonable assumption for measurements produced by a device such as an inertial guidance system because each parameter's error is a sum of several small errors. For this model the uncertainty regions are ellipses in the image plane. The derivation of this fact can be found in Appendix I.

Figure 7 shows a typical uncertainty ellipse that is prescribed to have a 95% probability of containing the actual occurrence of the feature. The 100 dots were produced by varying the camera parameters 100 different times according to the error model and by projecting the three-dimensional feature point onto the image plane containing the ellipse. Notice that 92 of the points are inside the ellipse, which is consistent with the 95% prediction.

Having found one feature, one would expect that its location would greatly restrict the possible locations for a nearby feature. This idea leads to a second type of uncertainty region, a relative uncertainty region. In addition to the normal information used to compute an uncertainty region, a relative uncertainty region is a function of another feature and its location. Since the location of a nearby feature typically adds constraints on the possible locations for a feature, the relative uncertainty region is usually significantly smaller than the regular uncertainty region. Given the assumption that the camera parameters vary according to a joint normal distribution, the relative uncertainty regions are also ellipses. A derivation of the mathematical description of a relative uncertainty region is given in Appendix II.

A relative uncertainty region is used to reduce the amount of work required to locate a second feature after a nearby feature has been found. This is particularly useful when a possible match for a feature is being verified. The logic is as follows: if this is feature A, then feature B should be in a small region over there; if B is not there, this must not be A.

Figure 8 shows the initial uncertainty ellipse and the relative uncertainty ellipse about a point feature. The large ellipse is the uncertainty region predicted from the uncertainties about the camera parameters. The small ellipse is the relative uncertainty region derived from the location of the arrow just above it in the picture.

POINT-ON-A-LINE MATCHES

Most people use point-to-point matches to refine correspondences. Since roads are the major objects of interest for the road expert, we wanted to include them as features that could be used within the image-to-database correspondence phase as well as in the monitoring phase.

There is a built-in trade-off between point features and line features, such as roads: it is easier to find a point on a line than it is to locate a point feature, but less information is gained by doing so. Point-to-point matches produce twice the number of constraints for the refinement process, but they are generally more expensive to find because an area search is required as opposed to a linear search for point-on-a-line matches.

To use linear features we needed an operator (or operators) to find points on roads and we had to to extend the correspondence refinement process to include the new type of feature match.

Point-on-a-Line Operators

Currently we have two operators that locate points on a road. One is used at low resolution (e.g., 20 foot/pixel) when roads appear as lines, and one is used at high resolution (e.g., 1 foot/pixel) when the internal structure of the road is discernable. The low resolution operator is an extension of the Duda road operator, which has been discussed in previous SRI image-understanding reports [2]. The high resolution operator is an adaptation of Quam's road tracking operator [12]. It performs a 1-D correlation of the expected road cross section to locate possible points on the road and then tries to track the road for a short distance to make sure that the candidate point is part of the expected road.

Correspondence Refinement

The correspondence refinement process (or "optimizer") is based on Gennery's approach to calibration [10]. It solves the nonlinear problem by iteratively solving linear approximations. For point-to-point matches a 3-D point in the world is matched with a 2-D point in the image. In that case the optimizer has two residuals per match to use to improve the camera parameter estimates: the X and Y components of the difference between the predicted image of the world point and the point in the image at which the operator located its match. If instead of locating a specific point, an operator locates a point on a line, the optimizer only has one residual to use because the point could be any place along the line. The residual for a point-on-a-line match is the distance from the point to the line. As the optimizer searches for improved camera parameters, the image of the 3-D line should get closer to the point located by the operator, but the closest point on the line may slip back and forth along the line.

So far the optimizer has only been extended to handle point-on-a-line matches. However, since roads are generally constructed as combinations of linear segments and arcs of circles, it may be useful to extend the optimizer to include other types of matches that involve a point and an analytic curve, e.g., a point-on-an-ellipse match. The main components of such an extension are (1) a procedure to compute the distance between a point and the curve and (2) a procedure to compute the partial derivatives of that distance with respect to the camera parameters.

The optimizer could even be extended to arbitrary curves by incorporating a procedure, such as chamfering [3], that computes the distance between a point and an arbitrary curve. Unfortunately, such distance computations are generally expensive.

The current implementation of the optimizer is relatively fast. It takes one second on our KL-10 to perform one iteration when 100 residuals are used to refine the estimates. (Recall that each point-to-point match adds two residuals; each point-on-a-line match adds one residual.) Five to ten iterations are normally required to achieve convergence, which is defined to be a state in which the parameter adjustments are on the order of .00005 units.

As Gennery points out, the optimizer can be used to filter out "mistakes" by iteratively deleting the match with the largest residual until the deletion no longer significantly improves that point's residual. In practice this heuristic has proven to be useful, but it is expensive and theoretically unsound. For example, consider Figure 9, which shows a set of points through which a line is to be fitted using a least-squares approach. The one "mistake" happens to draw the line toward it in such a way that the point with the worst residual after convergence is one of the "good" points. Deleting the point with the worst residual and trying again only repeats the situation. The conclusion is to try to filter out mistakes before they are given to the optimizer. The next section describes some of the ways this filtering or verification can be done.

FEATURE VERIFICATION

As mentioned in the last section, it appears to be more cost-effective to filter out mistakes, if at all possible, before applying the optimizer. We have identified four possible methods for performing such filtering:

(1) Operator threshold - Be suspicious of any match for which the operator does not produce a confidence above a certain threshold; e.g., if a 2-D correlation operator produces a correlation of less than .8, ignore its results.

(2) Self support - Be suspicious of any match that cannot be verified by locating a larger portion of the same feature; e.g., if an operator locates a point that is supposed to be on a road but the road tracker cannot extend the match, ignore it.

(3) Pairwise support - Be suspicious of any match that is not positioned correctly relative to some other feature that has already been located; e.g., if an operator locates an arrow on a road and its matching location is not at a reasonable distance from another nearby feature that has been verified, ignore the match.

(4) Group support - Be suspicious of any match that is not positioned correctly relative to a group of other features that have already been located, e.g., if three point features have been found and verified, ignore a match for a fourth feature that does not appear at the correct relative location.

We differentiate between these methods (or heuristics) because they generally require different models and techniques.

It is relatively straightforward to apply all of the verification methods to point features. The relative uncertainty regions can be used to determine if two features are mutually consistent. This pairwise consistency can be extended to group consistency through maximal clique techniques [1] or through optimal embedding techniques [7].

The extension to group consistency can be achieved by constructing a graph that has one node for each match and a link between each pair of nodes that is pairwise consistent. The largest completely connected subgraph (i.e., the largest maximal clique) represents the largest set of mutually consistent matches. Any match that is not in that set is pairwise inconsistent with at least one of the matches in the set. Thus, it is suspicious.

Additional care has to be taken to apply the verification techniques to point-on-a-line matches. The important test is to be able to distinguish pairwise consistent matches from pairwise inconsistent matches when one or more of the matches is a point-on-a-line match. Figure 10 shows the three significantly different cases. In Figure 10a one of the two matches is a point-to-point match and one is a point-on-a-line match. If the slope of the line is known accurately, the distance between the point and the line can be used to determine if the matches are consistent. Since the uncertainties associated with each camera parameter are relatively small, the slope of the line should remain relatively constant. Thus the distance from the point to the line should be relatively constant.

In Figure 10b both of the matches are point-on-a-line matches and the lines are essentially parallel. In this case the distance between the lines is sufficient to check the relative positions of the two matches. For example, if an operator is trying to locate both sets of lanes on a freeway, the distance between the two sets of lanes should be within a predetermined range.

If both of the matches are point-on-a-line matches and the lines are not parallel, as in Figure 10c, some additional information is needed in order to check their relative consistency. One solution is to intersect the two lines and use that point in conjunction with a third match to check the relative position of all three matches.

EXAMPLE

We have implemented one fixed strategy in terms of the verification techniques and are just beginning to explore the possibility of automatically tailoring the verification strategies to fit specific sets of features and tasks. The example task is to refine the image-to-database correspondence for the picture shown in Figure 4 using its full resolution of approximately 2 feet/pixel. The initial uncertainties about the camera parameters imply uncertainties in the image of plus or minus 95 pixels, which correspond to approximately plus or minus 190 feet on the ground. The goal is to reduce these uncertainties to approximately plus or minus one pixel, an increase in precision of almost two orders of magnitude.

The database used in this example contains two types of features, linear road segments and road surface markings. Figure 11 shows the features that are available for this site. The lines represent the road segments and the pluses represent the surface markings. The appearance of each road segment is described by a road cross section model. The appearance of a surface marking is described by an image patch from a previous picture of the site.

A fixed strategy has been implemented to use these features to perform the task and demonstrate our new techniques. The basic approach is to locate the linear features first because they are less expensive to find, use them to refine the camera parameters, locate the point features, use them to verify the first refinement, and then perform a second refinement using both the points and the lines.

Given estimates for the camera parameters, the system predicts the location of the road segments in the new picture. Figure 12 shows these predictions, which are shifted left and down approximately 60 pixels from their actual locations. The estimates of the camera parameters are also used to warp each road cross section to the expected size and orientation of the corresponding road segment. In addition, the estimates of the uncertainties about the camera parameters are used to predict the uncertainty regions about the center points of each linear segment. Figure 13 shows these uncertainty ellipses that have a 95% probability of containing the desired point.

The search strategy for a linear feature is to look along lines perpendicular to the expected location of the feature. The lengths of the lines are determined by the size of the uncertainty ellipse.

The high-resolution, one-dimensional correlation operator is applied along the search line to locate points that may be on the desired road. The self-support method is used to verify each candidate point. The road tracker tries to track the road for a short distance. If it cannot, the point is abandoned. Figure 14 shows an example of the application of self support. The line on the left is the predicted location of the road segment. The other line, which is crossed like a T, represents the location of the match and the results of the road tracker following the road.

For some road segments self-support is not sufficient to locate the desired road because there are two or three parallel roads that all look alike. In order to distinguish one road from another, preplanned groups of features have been established within which pairwise and group support can be obtained. For example, Figure 15 shows a set of three sets of lanes, two of which are difficult to tell apart simply by looking at their road cross sections. The relative locations of the three sets of lanes are used to determine the correct matches. The lines perpendicular to the roads indicate the final choice for a consistent set of matches.

Figure 16 shows the results of searching for all of the road segments in the database (shown in Figure 11). Two of the roads were not found because the contrasts were not sufficient to produce matches with the desired confidence. The matches were given to the optimizer along with the initial estimates of the camera parameters and the uncertainties about the estimates; the optimizer produced new estimates for the parameters and new uncertainties. Figure 17 shows the new predictions for the locations of the road segments. The new uncertainties imply uncertainties in the image of approximately plus or minus 1.5 pixels, close to our goal.

To verify the new estimates the surface markings were located. The new estimates were used to predict the locations and appearances of the features; the new uncertainties were used to predict the uncertainty regions; and two-dimensional correlation was used to locate the features. The average difference between the predicted location and the matching location was approximately 1.3 pixels and the largest distance was 1.7 pixels. The final refinement based on both the lines and the points reduced the uncertainties in the image to approximately 1.1 pixels, which is very close to our goal and corresponds to approximately 2.2 feet on the ground.

CONCLUSION

We have described and demonstrated a set of techniques to perform some of the subtasks required in an automatic system to refine image-to-database correspondences. In particular, we discussed techniques to compute uncertainty regions, techniques to incorporate point-on-a-line matches, and techniques to verify the results of operators. These techniques were combined to form a strategy, which we demonstrated in an example task.

Additional research is required on several other key subtasks required in an automatic system; for example, the selection of features and the tailoring of a strategy to different tasks. Other needs include better feature modeling, better operators to locate features over a wide range of viewing angles and conditions, and an alternative to least-squares optimization.

REFERENCES

1. A.P. Ambler et al., "A Versatile Computer-Controlled Assembly System," Proc. Third IJCAI, pp. 298-307 (August 1973).

2. H.G. Barrow, "Interactive Aids for Cartography and Photo Interpretation," Semiannual Technical Report, SRI Project 5300, SRI International, Menlo Park, California (November 1976).

3. H.G. Barrow et al., "Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching," Proc. Fifth IJCAI, pp. 659-663 (August 1977).

4. R.C. Bolles, "Verification Vision for Programmable Assembly," Proc. Fifth IJCAI, pp. 569-575 (August 1977).

5. B.L. Bullock et al., "Finding Structure in Outdoor Scenes," Hughes Research Report 498, (July 1976).

6. L. Davis, "Shape Matching Using Relaxation Techniques," TR-480, Computer Science Dept., University of Maryland, (September 1976).

7. M. Fischler and R. Elschlager, "The Representation and Matching of Pictorial Structures," IEEE Trans. Comp., No. 22, pp. 67-92 (1973).

8. M. Fischler et al., "The SRI Road Expert: An Overview," Proc. Image Understanding Workshop (November 1978).

9. T.D. Garvey, "Perceptual Strategies for Purposive Vision," Technical Note 117, SRI International, Menlo Park, California (September 1976).

10. D.B. Gennery, "A Stereo Vision System for an Autonomous Vehicle," Proc. Fifth IJCAI pp. 576-582 (August 1977).

11. B.K.P. Horn and B.L. Bachman, "Using Synthetic Images to Register Real Images with Surface Models," Proc. Image Understanding Workshop, pp. 75-95 (October 1977).

12. L.H. Quam, "Road Tracking and Anomaly Detection in Aerial Imagery," Proc. Image Understanding Workshop, pp. 51-55 (May 1978).

13. B. Widrow, "The 'Rubber Mask' Technique," Pattern Recognition, Vol. 5, pp. 175-211 (1973).

APPENDIX I

A LINEAR MODEL FOR PREDICTING THE DISTRIBUTION OF ERRORS UNDER A PROJECTIVE TRANSFORMATION

Problem Statement

GIVEN the set of camera parameters {yi} which define a projective transformation from 3-space to a 2-dimensional image plane {xi}, i=1,2; and assuming that the {yi}, i=1,2,...J, are jointly distributed according to a multivariate normal distribution function with given covariance matrix M, THEN we wish to find a region in the image plane, centered about the point provided by the projective transformation H{yi}, which will be large enough to contain the image of the corresponding 3-space point to some given level of probability.

Linear Approximation

As an approximation to the way in which the errors in the camera parameters produce displacements of a projected point, we will assume that:

$$\Delta x_1 = \sum_j \left(\frac{\delta x_1}{\delta y_j} * \Delta y_j \right)$$

[1] and

$$\Delta x_2 = \sum_j \left(\frac{\delta x_2}{\delta y_j} * \Delta y_j \right).$$

The partial derivatives in the above equations can be computed from the projective transformation H or measured experimentally. The two linear equations can be represented in matrix notation as:

[2] $$\Delta x = T(\Delta y)$$

where the transform T is the 2 x J matrix of the partial derivatives of the xi with respect to the yj, over the J camera parameters.

To simplify our notation, we will assume that the image plane and 3-space coordinate axes have their origins at the projected and nominally imaged points, respectively. Thus, the deltas in equation [2] can be dispensed with.

The Error Model

The multivariate normal probability density function has the form (for dimensionality "n"):

[3] $$P(x|u,M) = \frac{e^{(-.5*(x-u)^T M^{-1}(x-u))}}{(2*\pi)^{\left(\frac{n}{2}\right)} * \sqrt{|M|}}$$

where: U=E{X}
M=E{(X-U)(X-U)^T}
|A|= determinant of A.

The covariance matrix M must be positive semidefinite. That is, for any n-dimensional vector Z with real components we have:

[4] $$Z^T MZ >= 0.$$

Theorem 1 [Ref. 1, pg. 25]:

> If Y is distributed according to [3] with mean vector U and covariance matrix M then:
>
> If X=TY+B with T a constant matrix and B a constant vector, then X is normally distributed with mean V=TU+B and covariance matrix W=E[(X-V)(X-V)^T]=TMT^T.

Thus, given our previously stated assumptions, we can now assert that the error distribution in the image plane will be a bivariate normal probability density function, having the same form as equation [3], but with mean vector V, and covariance matrix W, obtained as described in the above theorem.

In more explicit form we have:

[5] $$P(x_1,x_2|0,0,\rho,s_1,s_2) = \frac{e^{\left(-\frac{G}{2}\right)}}{2*\pi*s_1*s_2*\sqrt{1-\rho^2}}$$

where:

$$G = \frac{\left(\frac{x_1^2}{s_1^2} - \frac{2*\rho*x_1*x_2}{s_1*s_2} + \frac{x_2^2}{s_2^2}\right)}{(1-\rho^2)}$$

$$s_1 = \sqrt{E\{x_1^2\}} \qquad s_2 = \sqrt{E\{x_2^2\}}$$

$$\rho = E\left\{\frac{x_1*x_2}{s_1*s_2}\right\}.$$

We note that ρ is the coefficient of correlation between x1 and x2 and $(-1 \leqslant \rho \leqslant 1)$.

The contours of constant probability density in the image {x1,x2} plane are the loci where the exponent of the density function is constant. They are similar coaxial ellipses, with their axes parallel to the eigenvectors of the covariance matrix W. In particular, the major axis of the ellipse will make an angle of

[6] $$\alpha = \frac{1}{2} * ARCTAN\left(\frac{2*\rho*s_1*s_2}{(s_1^2 - s_2^2)}\right)$$

with the x1 axis.

To simplify our derivation of the dimensions of the ellipse needed to provide a given level of probability of containing the image of the 3-space point being projected, we will transform our coordinate axes in the image plane so that they lie along the major and minor axes of the coaxial constant probability ellipses. The resulting covariance matrix Q has the form:

[7] $$Q = \begin{pmatrix} q_1^2 & 0 \\ 0 & q_2^2 \end{pmatrix}$$

where the qi (the new variances) are the eigenvalues of the covariance matrix W. These eigenvalues are found by solving the following equation:

[8] $$0 = \begin{vmatrix} (s_1^2-q^2) & (\rho*s_1*s_2) \\ (\rho*s_1*s_2) & (s_2^2-q^2) \end{vmatrix}.$$

The resulting solutions are:

$$q_1^2 = \frac{1}{2} * \left((s_1^2+s_2^2) + \sqrt{(s_1^2-s_2^2)^2+4*\rho*s_1^2*s_2^2}\right)$$

[9] and

$$q_2^2 = \frac{1}{2} * \left((s_1^2+s_2^2) - \sqrt{(s_1^2-s_2^2)^2+4*\rho*s_1^2*s_2^2}\right).$$

Substituting q1^2 for q^2 in either of the two homogeneous equations in:

[10] $$0 = (W - q^2*I)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

allows us to solve for the ratio of the x1 to x2 coefficient in the major eigenvector and determine its angle with the x1 axis to be:

[11] $$TAN(\alpha) = \frac{(q_1^2 - s_1^2)}{(\rho*s_1*s_2)}.$$

The above expression can be simplified using the identity ARCTAN(A)=2*ARCTAN({SQRT[1+A^2]-1}/A) to give the result in [6]. In terms of covariance matrix Q, the bivariate normal density function has the form:

[12]

$$P(z_1,z_2) = \frac{e^{\left(-\frac{G}{2}\right)}}{2*\pi*q_1*q_2}$$

where: $G = \frac{z_1^2}{q_1^2} + \frac{z_2^2}{q_2^2}$.

The locus of $G=c^2$, where c is a constant is an equi-probability ellipse with major radius of length c*q1 and minor radius of length c*q2.

The area contained within this ellipse is c^2*q1*q2*PI and the differential area is 2*c*q1*q2*PI*Δc.

Thus, the probability p'' that the image of the nominally projected 3-space point will fall into the elliptic ring formed by the ellipses with parameters c and c+Δc is:

[13] $$p'' = c*e^{-\frac{c^2}{2}}*\Delta c.$$

Integrating p'' from 0 to c we get:

[14] $$P = 1 - e^{-\frac{c^2}{2}}$$

where P is the probability that the image of the nominally projected 3-space point will fall into the ellipse with parameter c (i.e., the ellipse with major axis of length c*q1, minor radius of length c*q2, and orientation of the major axis of B; see equations [6] and [9] for the values of q1,q2, and α).

Some typical values for P are:

	P	c
	.50	1.177
[15]	.90	2.146
	.95	2.448
	.99	3.035

We note that if s1=s2=s, and ρ=0, then q1=q2=s; the resulting contours are circles, and the parameter c corresponds to the radius of the resulting error circle measured in standard deviations (s). For this case, the radius which results in a 50% error probability is 1.177s, but the expected radial error is s*SQRT(PI/2)=1.253s, and the expected value of the square of the radial error is $E\{x1^2\}+E\{x2^2\} = 2*s^2$.

Finally, by invoking Bayes' theorem, we note that if an "error ellipse" as determined above is centered on the true projection of a given 3-space point, and has probability P of containing the actual projection of that point, then the same ellipse centered on the actual projection would have the same probability P of containing the true projection (assuming there is no difference in the way the true and actual projected points are distributed over the image plane).

Reference

1. T.W. Anderson, An Introduction to Multivariate Analysis (John Wiley & Sons, New York, New York, (1958).

APPENDIX II

RELATIVE UNCERTAINTY REGIONS

Let p and q be two three-dimensional feature points. Let a1 represent an estimate of the camera parameters. Let F represent the perspective transformation, which is a function of the camera parameters, that maps feature points into image points. Then

[1] P = F(a1,p) and Q = F(a1,q),

where P and Q are the 2-dimensional image coordinates of the points p and q. P and Q are the predicted image locations for the two features based on the estimates a1.

If an operator has correctly located the image of p at P', where should the image of q be? Or, in which region should the image of q appear? That is, what is the relative uncertainty region about q with respect to p and P'?

Assume that the actual camera parameters are a2 and the two features actually appear at P' and Q' in the image. Thus,

[2] P' = F(a2,p) and Q' = F(a2,q).

The relative uncertainty region can be described by the difference between (Q' - P') and (Q - P) as a function of a1 and a2.

Let

[3] a2 = a1 + Δa.

If we make the same assumption made in appendix I that the parameter space is locally linear about a1 and a2, then

[4] P' = F(a1,p) + Mp *Δa

and

[5] Q' = F(a1,q) + Mq *Δa

where Mp and Mq are the 2 x N matrices of partial derivatives that describe the relative changes in the image plane as a function of the N camera parameters. Then

[6] [(Q' - P') - (Q - P)] = Mq * Δa - Mp *Δa

or

[7] [(Q' - P') - (Q - P)] = (Mq - Mp) * Δa.

If the Δa's are distributed according to a multivariate normal distribution, Theorem 1 in Appendix I applies. If the mean of the distribution is the vector U and the covariance matrix is S, the vectors on the left side of linear equation [7] will be distributed with mean V = (Mq-Mp)*U and covariance matrix W = $(Mq-Mp)*S*(Mq-Mp)^T$.

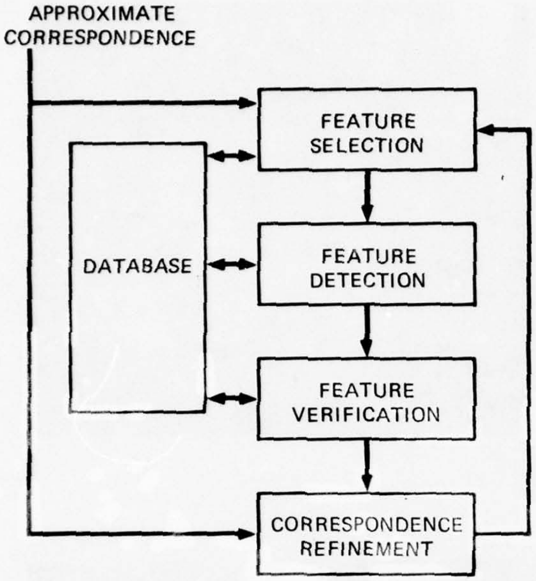


Figure 1.

GENERAL ASSUMPTIONS

(1) Road pictures
(2) Repetitive coverage
(3) Ground resolutions between
20 feet/pixel and 1 foot/pixel
(4) Database of roads and
other features
(5) Different sun angles
(6) Database features may be obscured
by clouds, terrain features, etc.
(7) Wide range of viewpoints
(8) Correspondence is a
perspective transformation
(9) Small parameter uncertainties
(10) Maximum uncertainty regions
on the ground of +-200 feet

INFORMATION FOR EACH IMAGE

(1) Internal camera parameters
(estimates & uncertainties)
(2) External camera parameters
(estimates & uncertainties)
(3) time of day and day of year
image was taken

Figure 2.

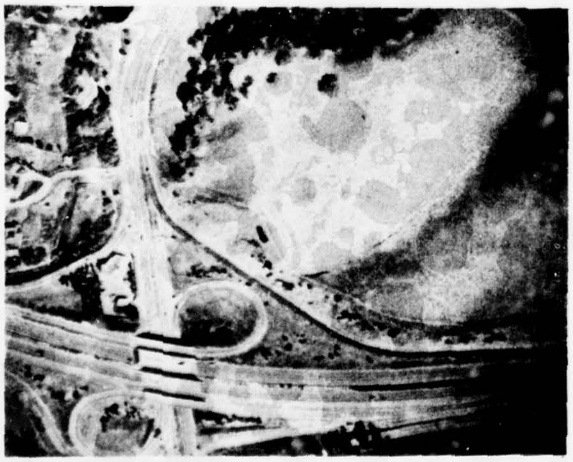


Figure 3.

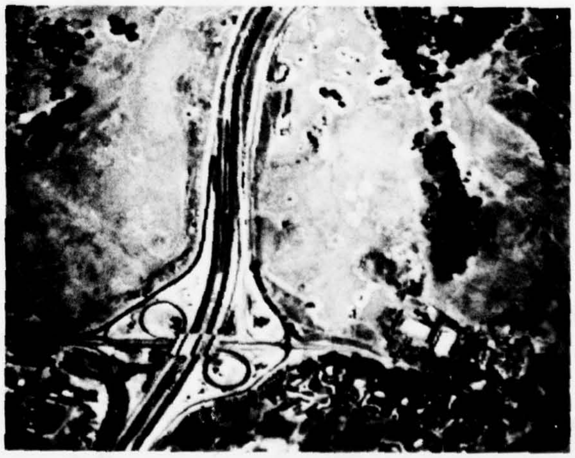


Figure 4.

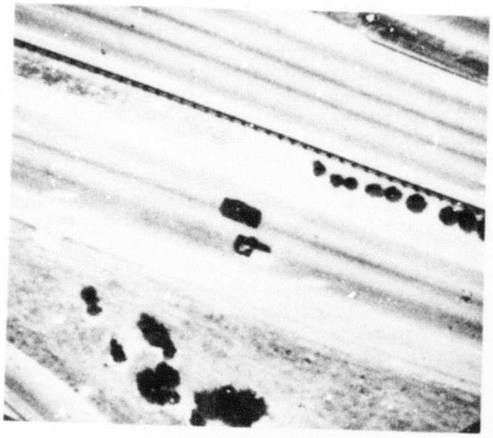


Figure 5.

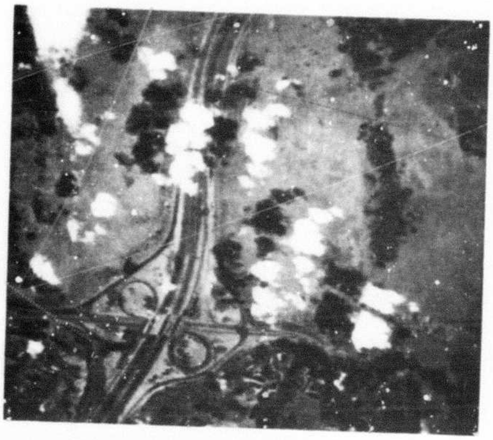


Figure 6.

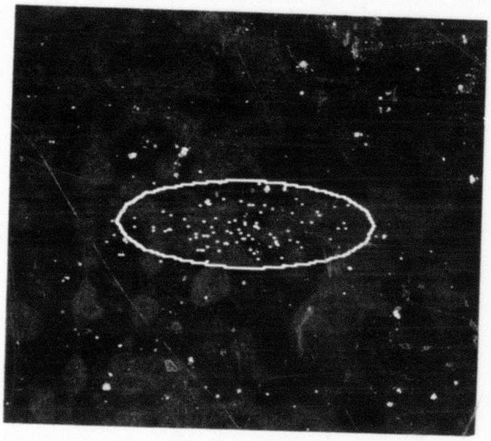


Figure 7.

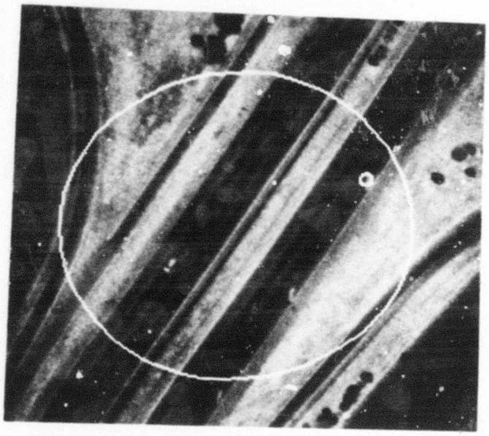


Figure 8.

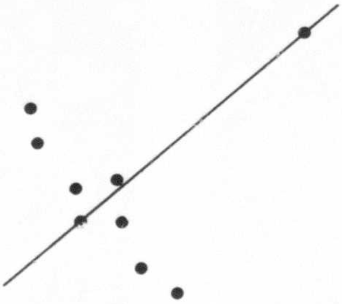


Figure 9.

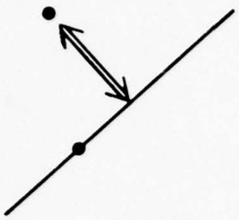


Figure 10a.

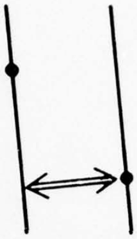


Figure 10b.

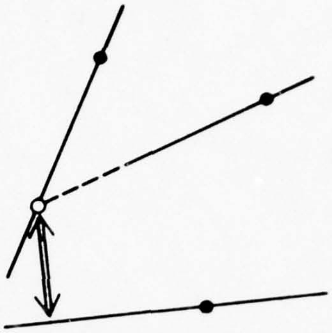


Figure 10c.

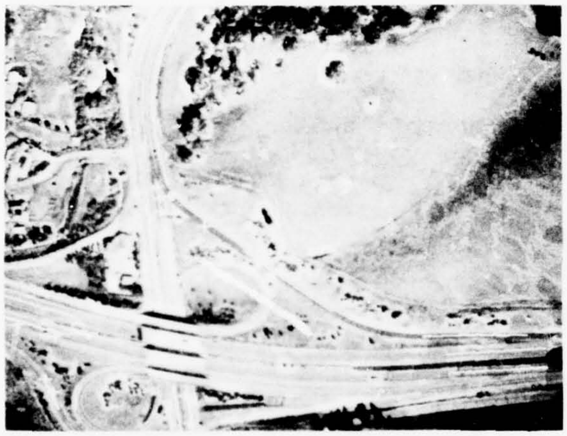


Figure 11.

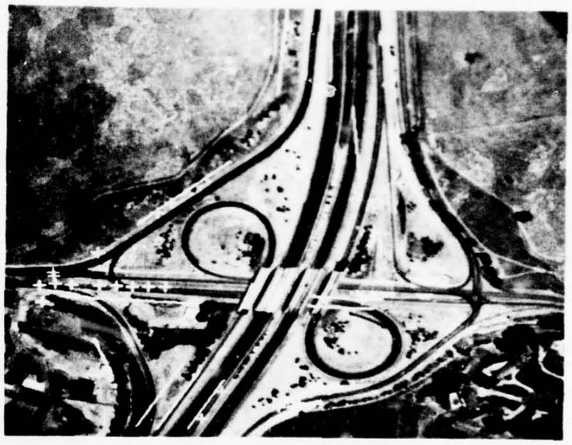


Figure 12.

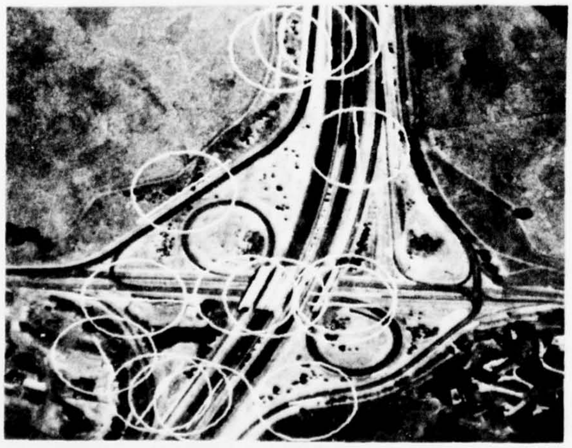


Figure 13.

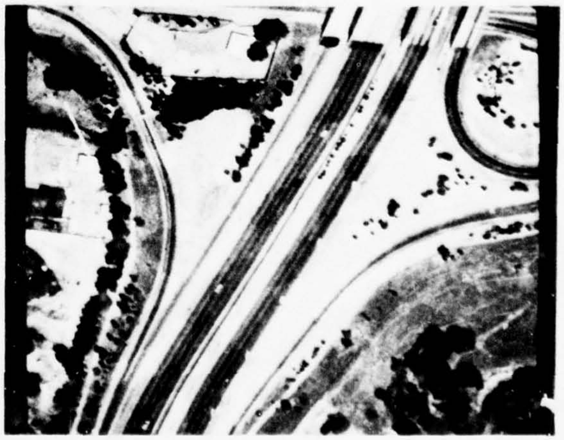


Figure 14.

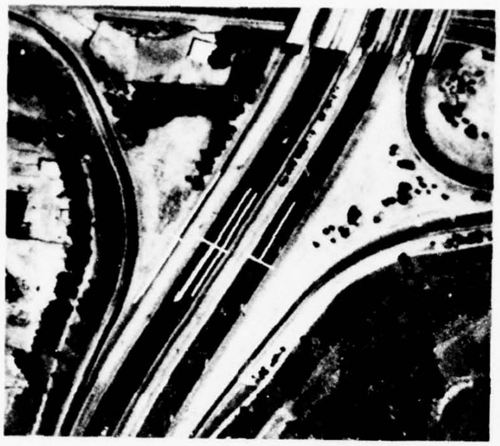


Figure 15.

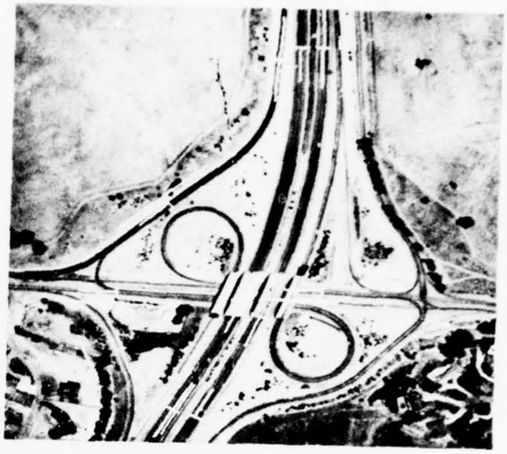


Figure 16.

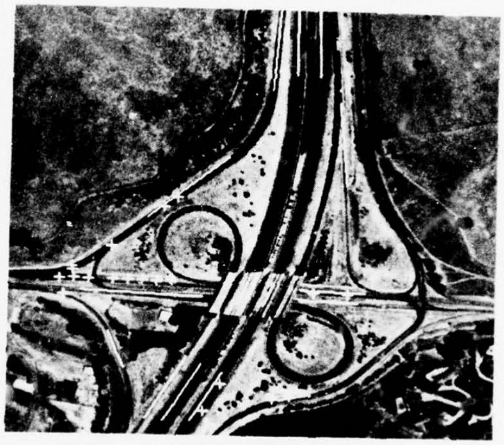


Figure 17.

SESSION V

HARDWARE

HARDWARE IMPLEMENTATION OF IMAGE PROCESSING USING OVERLAYS: RELAXATION

Thomas J. Willett

Westinghouse Systems Development Division, Baltimore, Maryland 21203

ABSTRACT

Under contract to University of Maryland, Westinghouse has been implementing algorithms for the image understanding process. The program is sponsored by DARPA and monitored by the Army's Night Vision Laboratory. Our objective is the examination of the latest advances in bit sliced microprocessor technology and the design of innovative architectures which are highly parallel, high speed fault tolerant, and require both a small instruction set and a small area.

INTRODUCTION

We first examine the discrete relaxation algorithm as described by the University of Maryland[1] and shcw two implementations in LISP. The non-linear probabilistic relaxation algorithm is then considered and an implementation in bit sliced microprocessors is discussed using a single instruction-multiple data architecture.

DISCRETE RELAXATION

Relaxation is essentially an iterative technique where the relationships between objects are used to classify them; specific image characteristics (objects) are used to classify the image figures. For example, the objects could be line segments detected in the image. If there were four of them and they were at right angles, one might conclude that they formed some sort of a rectangular figure. Objects can also be other image characteristics such as blobs, straight lines, or junctures which, by themselves, do not have much meaning. But considered together, the classification of the figure becomes apparent. We examine the relationships for consistency, i.e., if the objects form a particular figure (rectangle), they must have a certain relationship to each other for each part of the figure. Further, inconsistent relationships must be rejected. In a deeper cut through the problem, we may perform the iteration to find a consistent classification by discarding inconsistent relationships. To show how this is done, let us return to our previous example. Suppose the four objects have several possible relationships between pairs, and we are considering the relationships at the pairwise level only. One way to iterate is to assume a certain classification for object number 1 and cycle through the relationship between object 1 and each of the other objects in parallel. If the classification of object number 1 is inconsistent with one of the other objects, the classification for object number 1 is rejected and the next classification is tried. Clearly, the analyst can end up with a set of consistent classifications, none of which dominates. This is the shortcoming of the discrete case and is handled in the probabilistic approach. The next item of interest is how the relationships are examined for consistency, as outlined in the Maryland paper[1].

Assume there are three objects a_1, a_2, and a_3, and their possible classification can be λ and μ. More specifically, a kind of graph can be formed as shown in figure 1. The dots show that the objects can be represented as a λ or μ. If it were not possible, e.g., to represent a_3 as a μ, there would be no dot at the (μ, a_3) position. Suppose, further that the following arbitrary set of relationshirs exist between the objects:

$$A = A_1 = A_2 = A_3 = \{\lambda,\mu\} \quad (1)$$

$$A_{12} = A_{23} = \{(\lambda,\lambda),\ (\mu,\mu)\} \quad (2)$$

$$A_{13} = \{(\lambda,\mu),\ (\mu,\lambda)\} \quad (3)$$

(1) states that the objects a_1, a_2, a_3 can be represented as either λ or μ, i.e., $\{\lambda,\mu\}$.

(2) states that the relationship between a_1 and a_2 is the same as that between a_2 and a_3 and can be characterized as λ for each or μ for each.

(3) states that the relationship between objects a_1 and a_3 can be stated as either λ for a_1 and μ for a_2 or μ for a_1 and λ for a_2. Then these relationships can be drawn as arcs on the graph as shown in figure 2. Now, we see from figure 2c that there is an arc between each of the objects which symbolizes the idea that there should be a consistent relationship among them. However, if we trace our way around the graph we find that $a_1 = \lambda$, $a_2 = \lambda$, and $a_3 = \lambda$, but to return to a_1 means that $a_1 = \mu$ -- a contradiction. On the other hand, the graph of figure 3 represents a case when there are two consistent and possible interpretations of the set of relations. The two consistent classifications from figure 3 are (λ,λ,μ) and (μ,μ,λ) for objects a_1, a_2, a_3, respectively. Next, we consider hardware implementation.

To form the graphs on a digital machine, we assume a_1 is classified as λ. We cycle classifications for a_2 and a_3 against it by matching λ's. So for A_2, we obtain (λ,λ) and for a_3 we obtain (λ,μ). Then for a_1, a_2, a_3 we obtain (λ,λ,μ). Similarly, assuming $a_1 = \mu$, we obtain (μ,μ,λ). These are the same two consistent classifications shown in the graph of figure 3.

Since we are manipulating lists of symbols rather than lists of numbers, a natural computer language for this problem is LISP[2]. Referring to LISP, we note that some defined functions are directly applicable to the problem. First of all, there are two possibilities for a_1, λ or μ. This should be compared with the first of each two-tuple of A_{12}, i.e., λ of (λ,λ) or μ of (μ,μ) which represents a_1. In the language of LISP, $A_1 = (\lambda\cdot\mu)$, $A_{12} = [(\lambda\cdot\lambda)\cdot(\mu\cdot\mu)]$, $A_{13} = [(\lambda\cdot\mu)\cdot(\mu\cdot\mu)]$. Further, to make the form of A_1 compatible with that of A_{12} and not change the meaning of A_1, we let $A_1 = [(\lambda\cdot\mu)\cdot(\lambda\cdot\mu)]$. Then we could employ the pairlis and equal functions sequentially. The function pairlis [x; y; a] gives the list of pairs of corresponding elements of the lists x and y, and appends them to the list a. As an example, let x = (X1•(X2•X3)) and y = (Y1•(Y2•Y3)) which are to be paired and added to a list (X4•Y4)•(X5•Y5), then pairlis[X; Y; a] = (X1•Y1)•(X2•Y2)•(X3•Y3) •(X4•Y4)•(X5•Y5). Then pairlis $[A_1;\ A_{12};\ \ell]$ $= (\lambda\cdot\lambda)\cdot(\mu\cdot\lambda)\cdot(\lambda\cdot\mu)\cdot(\mu\cdot\mu)$

and equal $(\lambda\cdot\lambda)$ = True;

we obtain λ for a_2 from the second pair, assuming we remembered a_2's position in that pair. Simultaneously, pairlis $[A_1;\ A_{13};\ \ell]$ is computed to obtain the classification for a_3. A more direct approach in LISP is the "SASSOC" function which has the following definition:

sassoc [x; y; μ]: searches y, which is a list of dotted pairs for a pair whose first element is x. If such a pair is found, the value of sassoc, μ, is this pair.

```
sassoc [x; y; μ] = [null [y] → μ[ ];
                    eq [caar [y]; x] → car [y];
                    T → sassoc [x; cdr [y]; μ]].
```

Applying sassoc,

$$A_1 = [\lambda,\mu],\ \text{car}\ [A_1] = \lambda = x$$
$$y = A_{12} = [(\lambda\cdot\lambda)\cdot(\mu\cdot\mu)]$$

then,

```
sassoc [λ; (λ•λ)•(μ•μ)); μ] =
  Step 1.
        null [y] is false
        caar [(λ•λ)•(μ•μ)] = car [λ•λ] = λ
        eq [carr [y]; x] = eq [λ;λ] = T
         •
        • •
         sassoc = car [y] = (λ•λ).
```

The pair has been found; the second atomic symbol of the S expression is the classificaiton for a_2, namely λ. Now, we repeat sassoc for A_{13} to find a consistent classification for a_3.

$$A_1 = [\lambda,\mu],\ \text{car } [A_1] = \lambda = x$$
$$A_{13} = [(\lambda\cdot\mu)\cdot(\mu\cdot\lambda)] = y$$

then

```
    sassoc [λ, ((λ•μ)•(μ•μ); μ] =
      Step 1.
             null [y] is false
             caar [(λ•μ)•(μ•μ)] = car [λ•μ] = λ
             eq [caar [y]; x] = eq [λ•λ] = T
             .
             .
             .
             sassoc = car (y) = (λ•μ).
```

and the classification for a_3 is the second atomic symbol in the S expression, i.e., μ for a_3, and the classification becomes (λ,λ,μ) for (a_1, a_2, a_3). We would then repeat the procedure above where a_1 starts with μ, and we obtain (μ,μ,λ) for (a_1, a_2, a_3).

In summary, we have shown that there are at least two LISP structures which produce the consistent classification lists for objects a_1, a_2, a_3 as also shown in the graph of figure 3.

In terms of bit slice implementation, we might assign a processor to each object. The width of each processor would be the width of the classification word. It is worth pointing out that, since each processor would be doing the same thing, it is possible to have one controller for all three CPU's. This reduction in hardware is not possible with microprocessors which are not bit sliced.

NON-LINEAR PROBABILISTIC RELAXATION

A more realistic approach to relaxation is the non-linear probabilistic one[1]; here probabilities are assigned to denote the possibility of an object being in a particular class. The strategy is to enforce the probability $p_i(\lambda)$ of a given class of a given object a_i if other objects' labels, having high probabilities, are highly compatible with λ at a_i. On the other hand, $p_i(\lambda)$ should be decreased if other high probability labels are incompatible with λ at a_i. Further, low probability labels should have little effect on $p_i(\lambda)$ regardless of whether they are compatible with it. We may then set up the matrix of $p_i(\lambda_\ell)$'s and iterate the matrix by some function G, $p_i^{(k+1)}(\lambda) = G = G[p_i^k(\lambda),\ r_{ij}(\lambda\lambda'),\ d_{ij}]$, according to the above strategy. The quantities $r_{ij}(\lambda,\lambda')$ and d_{ij} are the compatibility coefficients and weights, respectively. For our implementation, we shall assume the $r_{ij}(\lambda,\lambda')$'s are the statistical correlation coefficients between object a_i, class λ and object a_j, class λ'.

The function $q_i^k(\lambda) = \sum_j d_{ij}\ [\sum_{\lambda'} r_{ij}(\lambda\lambda')\ q_j^k(\lambda')]$ has properties which follow the above strategy, i.e., if $p_j^k(\lambda')$ is high, and $r_{ij}(\lambda\lambda')$ is highly positive or negative, then $q_j^k(\lambda)$ reflects this. However, a small $p_j^k(\lambda')$ makes a relatively smaller contribution regardless of $r_{ij}(\lambda\lambda')$. In order to ensure that $p_i^{k+1}(\lambda)$ is non-negative, and that $\sum_\lambda p_i^{k+1}(\lambda) = 1$, we define

$$p_i^{k+1}(\lambda) = p_i^k(\lambda)[1 + q_i^k(\lambda)]/\sum_\lambda p_i^k(\lambda)(1 + q_i^k(\lambda)]$$

and again the strategy is obeyed.

For hardware implementation, we concentrate on q_i^k and expand it for the simple case of two classes $\lambda = \lambda_1, \lambda_2$ and three objects i = 1, 2, 3 as shown in figure 4. We note several possible simplifications in the expansion, namely $r_{ii}(\lambda\lambda) = 1$ and $r_{ij}(\lambda\lambda') = r_{ji}(\lambda'\lambda)$. Replacing each of the correlation coefficients by capital letters A, B, ...0, i.e., $A = r_{11}(\lambda_1, \lambda_2) = r_{11}(\lambda_2,\lambda_1)$, $B = r_{21}(\lambda_1\lambda_1) = r_12(\lambda_1\lambda_1)$ ..., we can write $q_i^k(\lambda)$ as shown in figure 5. Row 1 of each expression is composed of d_{11}, A, $p_1^0(\lambda_1)$ and $p_1^0(\lambda_2)$; the only difference is the relative position of the A coefficient. Similarly, rows 5 and 9 have the same structure. The same kind of remarks can be made about the other row pairs, e.g., 2 and 4. In fact, figure 6 shows the similarities. Because of the similarities in structure between rows, the same set of microinstructions, including rotation and register index, can form each side of each row. For example, consider a temporary storage and instruction set shown in figure 7 for rows 2 and 4. With 30 microinstructions, rows 2 and 4 of figure 5 can be formed. Now consider the array shown in figure 8, where all the rows of $q_i^k(\lambda)$ are formed.

The boxes may be considered as each comprising a bit sliced ALU, AMD 2901/03, each of which is four bits wide. The register set shown previously is a RAM stack 16×4 atop each ALU and part of the ALU monolithic chip. The instruction set controls all four ALU's in parallel. Since the data is in immediate memory, cycle times are of the order of 200 nanoseconds or less. Hence, 30 microinstructions can be executed in 6 microseconds. Done in the parallel fashion as described above, $q_i^k(\lambda)$ can be computed in 6 microseconds using four bit-sliced ALU's and one controller.

In summary, we have shown the beginnings of applying bit sliced microprocessors to the relaxation algorithm. An array of four AMD 2900/03 ALU's can compute an intermediate quantity, $q_i^k(\lambda)$, of relaxation in approximately 6 microseconds, with a single instruction set for all four ALU's. Computation for an entire iteration of the two objects by three classes case is probably 8 microseconds or so, and ten iterations could be accomplished in approximately 100 microseconds. We have done this by taking advantage of certain symmetries in the calculations which hold in a number of real cases.

In the next period, we shall be expanding the basic problem to 10 classes and 100 objects and begin considering the interconnect problem and dynamic reconfiguration for a variable number of classes and objects, and reliability. It is important to consider special implementation for relaxation operations in order to provide for real or non-real time operations. The University of Maryland has estimated that it will require many hours to perform the relaxation computations for one image frame on a general purpose machine.

REFERENCES

1. Rosenfeld, A., Hummel, R.A., Zucker, S.W., "Scene Labeling by Relaxation Operations," IEE Transactions, Vo. SMC-6, No. 6, June 1976.

2. McCarthy, et al, LISP Programming Manual 1.5, MIT Press, 1962.

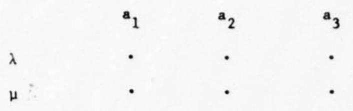


Figure 1. Graph Form of Relaxation

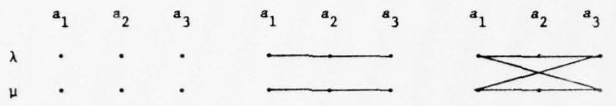


Figure 2a. Relationship (1)

Figure 2b. Relationship (1) + (2)

Figure 2c. Relationship (1) + (2) + (3)

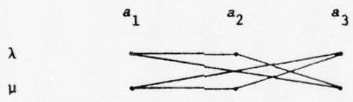


Figure 3. Two Consistent and Possible Situations

$$q_1^0(\lambda_1) = d_{11}[r_{11}(\lambda_1\lambda_1)\ p_1^0(\lambda_1) + r_{11}(\lambda_1\lambda_2)\ p_1^0(\lambda_2)] \qquad (1)$$
$$+\ d_{12}[r_{12}(\lambda_1\lambda_1)\ p_2^0(\lambda_1) + r_{12}(\lambda_1\lambda_2)\ p_2^0(\lambda_2)] \qquad (2)$$
$$+\ d_{13}]r_{13}(\lambda_1\lambda_1)\ p_3^0(\lambda_1) + r_{13}(\lambda_1\lambda_2)_3p^0(\lambda_2)] \qquad (3)$$

$$q_1^0(\lambda_2) = d_{11}[r_{11}(\lambda_2\lambda_1)\ p_1^0(\lambda_1) + r_{11}(\lambda_2\lambda_2)\ p_1^0(\lambda_2)]$$
$$+ d_{12}[r_{12}(\lambda_2\lambda_1)\ p_2^0(\lambda_1) + r_{12}(\lambda_2\lambda_2)\ p_2^0\ (\lambda_2)]$$
$$+ d_{13}[r_{13}(\lambda_2\lambda_1)\ p_3^0(\lambda_1) + r_{13}(\lambda_2\lambda_2)\ p_3(\lambda_2)]$$

$$q_2^0(\lambda_1) = d_{21}[r_{21}(\lambda_1\lambda_1)\ p_1^0(\lambda_1) + r_2(\lambda_1\lambda_2)\ p_1^0(\lambda_2)] \qquad (4)$$
$$+\ d_{22}[r_{22}(\lambda_1\lambda_1)\ p_2^0(\lambda_1) + r_{22}(\lambda_1\lambda_2)\ p_2^0(\lambda_2)] \qquad (5)$$
$$+\ d_{23}[r_{23}(\lambda_1\lambda_1)\ p_2^0(\lambda_1) + r_{23}(\lambda_1\lambda\ )_2p_2^0(\lambda_2)] \qquad (6)$$

$$q^0_2(\lambda_2) = d_{21}[r_{21}(\lambda_2\lambda_1)\ p_1^0(\lambda_1) + r_{21}(\lambda_2\lambda_2)\ p_1^0(\lambda_2)]$$
$$+\ d_{22}[r_{22}(\lambda_2\lambda_1)\ p_2^0(\lambda_1) + r_{22}(\lambda_2\lambda_2)\ p_2^0(\lambda_2)]$$
$$+\ d_{23}[r_{23}(\lambda_2\lambda_1)\ p_3^0(\lambda_1) + r_{23}(\lambda_2\lambda_2)\ p_3^0(\lambda_2)]$$

$$q_3^0(\lambda_1) = d_{31}[r_{31}(\lambda_1\lambda_1)\ p_1^0(\lambda_1) + r_{31}(\lambda_1\lambda_2)\ p_1^0(\lambda_2)] \qquad (7)$$
$$d_{32}(r_{32}(\lambda_1\lambda_1)\ p_2^0(\lambda_1) + r_{32}(\lambda_1\lambda_2)\ p_2^0(\lambda_2)] \qquad (8)$$
$$d_{33}[r_{33}(\lambda_i\lambda_1)\ p_3^0(\lambda_1) + r_{33}(\lambda_1\lambda_2)\ p_3^0(\lambda_2)] \qquad (9)$$

$$q_3^0(\lambda_2) = d_{31}[r_{31}(\lambda_2\lambda_1)\ p_1^0(\lambda_1) + r_{31}(\lambda_2\lambda_2)\ p_2^0(\lambda_2)]$$
$$+\ d_{32}[r_{32}(\lambda_2\lambda_1)\ p_2^0(\lambda_1) + r_{32}(\lambda_2\lambda_2)\ p_2^0(\lambda_2)]$$
$$+\ d_{33}[r_{33}(\lambda_2\lambda_1)\ p_3^0(\lambda_1) + r_{33}(\lambda_2\lambda_2)\ p_3^0(\lambda_2)]$$

Figure 4. $q_i^k(\lambda)$ **Expansion**

$$\begin{array}{llcll}
q_1^0(\lambda_1) = & d_{11}[p_1^0(\lambda_1) + A\ p_1^0(\lambda_2)] & (1) & q_1^0(\lambda_2) = & d_{11}[A\ p_1^0(\lambda_1) + p_1^0\ (\lambda_2)] \\
 & + d_{12}[B\ p_2^0(\lambda_1) + C\ p_2^0(\lambda_2)] & (2) & & + d_{12}[F\ p_2^0(\lambda_1) + D\ p_2^0(\lambda_2)] \\
 & + d_{13}[G\ p_3^0(\lambda_1) + I\ p_3^0(\lambda_2)] & (3) & & + d_{13}[H\ p_3^0(\lambda_1) + O\ p_3^0(\lambda_2)] \\
 & & & & \\
q_2^0(\lambda_1) = & d_{21}[B\ p_1^0(\lambda_1) + F\ p_1^0(\lambda_2)] & (4) & q_2^0(\lambda_2) = & d_{21}[C\ p_1^0(\lambda_1) + D\ p_1^0(\lambda_2)] \\
 & + d_{22}[p_2^0(\lambda_1) + E\ p_2^0(\lambda_2)] & (5) & & + d_{22}[E\ p_2^0(\lambda_1) + p_2^0(\lambda_2)] \\
 & + d_{23}[K\ p_3^0(\lambda_1) + N\ p_3^0(\lambda_2)] & (6) & & + d_{23}[M\ p_3^0(\lambda_1) + J\ p_3^0(\lambda_2)] \\
 & & & & \\
q_3^0(\lambda_1) = & d_{31}[G\ p_1^0(\lambda_1) + H\ p_1^0(\lambda_2)] & (7) & q_3^0(\lambda_2) = & d_{31}[I\ p_1^0(\lambda_1) + O\ p_2^0(\lambda_2)] \\
 & + d_{32}[K\ p_2^0(\lambda_1) + M\ p_2^0(\lambda_2)] & (8) & & + d_{32}[N\ p_2^0(\lambda_1) + J\ p_2^0(\lambda_2)] \\
 & + d_{33}\ [p_3^0(\lambda_1) + L\ p_3^0(\lambda_2)] & (9) & & + d_{33}[L\ p_3^0(\lambda_1) + p_3^0(\lambda_2)]
\end{array}$$

Figure 5. Collecting Terms

ROWS 1, 5, 9	A, E, L, $p_1^0(\lambda_1)$, $p_1^0(\lambda_2)$, $p_2^0(\lambda_1)$, $p_2^0(\lambda_2)$, $p_3^0(\lambda_1)$, $p_3^0(\lambda_2)$, d_{11}, d_{22}, d_{33}
ROWS 2, 4	D, F, C, B $p_2^0(\lambda_1)$, $p_2^0(\lambda_2)$, $p_1^0(\lambda_1)$, $p_1^0(\lambda_2)$, d_{12}, d_{21}
ROWS 3, 7	G, I, H, O $p_3^0(\lambda_1)$, $p_3^0(\lambda_2)$, $p_1^0(\lambda_1)$, $p_1^0(\lambda_2)$, d_{13}, d_{31}
ROWS 6, 8	K, M, N, J $p_3^0(\lambda_1)$, $p_3^0(\lambda_2)$, $p_2^0(\lambda_1)$, $p_2^0(\lambda_2)$, d_{23}, d_{32}

Figure 6. Row Similarities

R_8			R_{16}
R_7			R_{15}
R_6			R_{14}
R_5	$p_2^0(\lambda_2)$	$p_1^0(\lambda_2)$	R_{13}
R_4	$p_2^0(\lambda_1)$	$p_2^0(\lambda_2)$	R_{12}
R_3	d_{12}	d_{21}	R_{11}
R_2	D	F	R_{10}
R_1	B	C	R_9

move $R_9R_{12} \rightarrow R_6R_{14}$	$p_2^0(\lambda_1)\ p_2^0(\lambda_2)$
move $R_9R_{12} \rightarrow R_7R_{15}$	$p_2^0(\lambda_1)\ p_2^0(\lambda_2)$
mult $R_3R_6 \rightarrow R_6$	$d_{12}^2\ p_2^0(\lambda_1)^2$
mult $R_3R_{14} \rightarrow R_7$	$d_{12}\ p_2^0(\lambda_2)$
mult $R_3R_7 \rightarrow R_7$	$d_{12}\ p_2^0(\lambda_2)$
mult $R_3R_{15} \rightarrow R_{15}$	$d_{12}p_2^0(\lambda_1)$
mult $R_1R_6 \rightarrow R_6$	$B\ d_{12}p_2^0(\lambda_1)$
mult $R_9R_{14} \rightarrow R_{14}$	$C\ d_{12}p_2^0(\lambda_2)$
add $R_6R_{14} \rightarrow R_{14}$	$B\ d_{12}p_2^0(\lambda_1) + C\ d_{12}p_2^0(\lambda_2)$
Remove R_{14}	
mult $R_2R_7 \rightarrow R_7$	$D\ d_{12}p_2^0(\lambda_2)$
mult $R_{10}R_{15} \rightarrow R_{15}$	$F\ d_{12}p_2^0(\lambda_1)$
add $R_7R_{15} \rightarrow R_{15}$	
Remove R_{15}	

Figure 7. Storage and Instruction Set

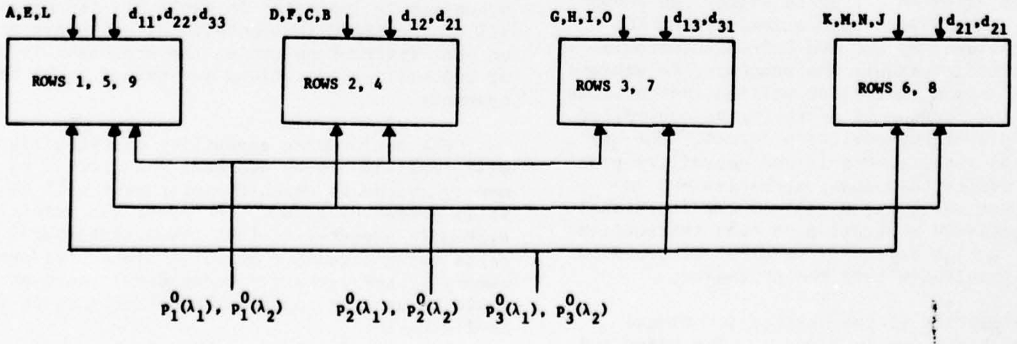


Figure 8. Processing array

SPARC - SYMBOLIC PROCESSING ALGORITHM RESEARCH COMPUTER

Gale R. Allen
Peter G. Juetten

Control Data Corporation
2800 East Old Shakopee Road
Minneapolis, Minnesota 55440

SUMMARY

This report summarizes progress on development of the SPARC Processor, which was started by ARPA in 1977. This is a joint research effort in computer architecture by Carnegie Mellon University and Control Data Corporation to develop a processor concept which anticipates future image processing needs. The research tasks are divided such that CMU is responsible for developing user system software aids and Control Data is responsible for the hardware development and basic operating system. Additional support is being provided by CDC for the development of more extensive system software and for construction of a second processor. By completion of this phase of the development in early Fall 1979, both CMU and CDC will have SPARC processors installed in their respective laboratories. At this time the processor design is approximately 75% complete and half the electronic parts have been placed on order. Cabinetry for the CMU and CDC processors has been ordered. Work is also in progress at CDC to develop a multiple processor architecture which utilizes extended versions of the SPARC processor. The key feature of this new architecture is a Ring System which is a flexible high-bandwidth interprocessor communication network.

ARCHITECTURE REVIEW

The SPARC processor consists of a number of functional units which communicate with each other via a generalized interconnection mechanism which can be thought of as an elaborate switch, as shown in Figure 1. A high performance semi-custom ECL technology, developed by CDC and Fairchild Corporation, is expected to enable the processor to achieve an instruction issue rate of 50 million instructions per second. The machine is microprogram controlled and has a single microinstruction format. The key feature of this architecture is the capability provided by the switch (crossbar) mechanism and instruction format which allows all of the functional units to be actively controlled on each instruction cycle. Thus, a high degree of parallel instruction execution is provided within the processor.

The capabilities of the various functional units in SPARC are given in Table 1. The adder and multiplier units can perform operations on 16-bit operands or can treat their inputs as two sets of 8-bit byte operands. In the case of the multiplier, this means that two independent 16-bit results would be produced simultaneously by two 8 x 8 bit multiply operations. A 32-bit result is produced when the operands are treated as 16-bit words. The adder and shift boolean units are capable of handling operands longer than 16 bits. In the case of SPARC, the two adder units can perform a 32-bit add or subtract on each instruction cycle.

A new functional unit has been defined which is called the Ring Port. The Ring Port allows multiple copies of the processor to be connected together by means of a Ring communications network. This is discussed later in the paper.

It is difficult to characterize the performance of the SPARC machine because of the parallelism in the SPARC architecture. Many functional units are operated on each instruction cycle and multiple operations can be performed in most of the units on each cycle. Some measured performance can be given, however, as indicated in Tables 2 and 3. The switch mechanism,described previously, provides a great deal of internal data transfer capability. Nearly all of the functional units can communicate with each other on each instruction cycle. The microinstruction word length is rather large, 200 bits, and is divided into 128 bits for functional unit control and 72 bits for control of the switch mechanism. The Ring Port accepts 16 bit word operands at the rate of 50 million per second. In addition, a memory port has been designed which has the capability to transfer 100 megabytes of data per second.

The maximum performance capability of the processor is indicated in Table 3. Assuming that all of the SPARC's computational units are active on each instruction cycle, the processor is capable of 200 million operations per second on 16 bit operands.

The performance capability described above will probably not be adequately supported by the memory system to which the processor will be initially connected at CMU. However, the initial system is expected to test the architectural concepts being developed here. A higher performance memory system currently under development at CDC could be used to upgrade the performance in the CMU configuration.

Other functional units can be developed to provide higher performance capabilities as may be required in certain applications. The processor is designed to allow new units to be connected to the machine without disturbing the physical hardware and without impact on the microinstruction format. Long-word, floating point and Fast Fourier Transform operations are examples of processes which may require specialized units in certain applications.

STATUS AND SCHEDULE

The objective of the current phase of this project is to produce a processor and have it installed and working at CMU early in the Fall of 1979, as shown in Figure 2. At this point, the hardware design is approximately 75% complete. The partitioning of SPARC functional units, including the adders, shift boolean unit, data memories, and multiplier into ECL LSI and MSI arrays has been completed and logic diagrams for these units have been prepared. Work is still in progress on the design of the control and input/output sections in which certain modifications are being implemented to simplify the hardware and enhance the utility of the processor from a software point of view. The gate level simulation of these units is also about 75% complete at this time.

The majority of the SPARC electronics hardware components consist of existing types of LSI ECL arrays which have been developed for future general-purpose CDC machines. However, in addition to using 14 existing array types, three new array types are being developed for SPARC. This development is being undertaken to improve the machine design, in particular to greatly reduce the chip count and improve performance. An example of where a new chip type is warrented is in the central data switching mechanism of the SPARC processor. This switch cannot be implemented with existing circuitry without a major sacrifice in machine capability. Since a new array type was necessary, the gate level design of the new array has been completed. An additional array which provides improved capability in the control section has been designed.

At present, more than one-half of the machine's electronic components have been ordered. The two new array types mentioned above have been placed on order with the CDC array development center. Also now on order is a processor cabinet which contains power supply wiring, power supplies, freon cooling condenser and freon plumbing and protection mechanisms. This cabinet,which has been recently developed by CDC for a new product line,has space for three full processors, and thus provides considerable expansion capability for future upgrades in hardware capability at CMU.

SPARC/PDP-11 INTERFACE

During this period a new approach was taken to interfacing SPARC to the PDP-11 equipment at CMU, as shown in Figure 3. With this design, the PDP-11 is provided with a Ring Port which has nearly the same capabilities as the Ring Port contained within the SPARC processor. Through this mechanism, the PDP-11 can communicate to any processor within a multiprocessor ring system.

SOFTWARE EFFORT

As indicated in the schedule, both CMU and CDC are engaged in developing software to support the SPARC processor. In the case of CMU, the microcode cross-assembler and register-level simulator will be designed to operate on the PDP-11 host computer for SPARC under the UNIX operating system. At CDC, the microcode cross-assembler and register-level simulator will be written in FORTRAN to operate on large CDC GP computers such as 6000, 7000,or CYBER 170. CDC will also develop a library of image understanding algorithms coded for SPARC and will analyze the performance of the hardware on these problems. CDC software also includes diagnostics and development of the basic operating system. The CDC software effort is being supported by Control Data and not by the Image Understanding Program.

Although the two microcode cross-assemblers will be implemented in different codes the basic user interaction features are expected to be equivalent. The basic design objectives for the assembler are listed in Table 4.

Work on providing a more usable instruction format for the programmer is continuing. The direction that the format development work is taking is indicated in Table 5, with examples of microinstructions shown in Table 6. In addition to the low level coding language, considerable work needs to be done on higher level languages. A language capability is needed to support initial algorithm development. As the algorithm matures portions of it (kernels) can be converted to higher performance microcode. In addition to these languages it appears that for many applications a FORTRAN programming capability will be needed.

APPLICATION REQUIREMENTS

There are a number of applications in which the performance capabilities of a single SPARC type processor will not meet the computational requirements. Typically on these applications the same algorithm or program is run continuously. The same algorithm or a small number of algorithms are repeatedly used to process the data. Examples of such applications are given in Table 7.

In addition to the need for high compute capability, such applications often require very large data base data storage in the form of a memory hierarchy. In some applications the input/output requirements can exceed several hundred megabits per second thus requiring a very flexible and high performance system I/O structure. Although the processing system tends to be dedicated in these applications, there is a need to be able to run several different types of algorithms on the same processor array. Therefore, there is the need for a reconfigurable structure and in general, specialized configurations are to be avoided. High reliability is required and single-point failure mechanisms must be avoided. There must also be a capability for on-line replacement of failed equipment so as to be able to continue procesing even as portions of the systen fail and aie replaced.

An interprocessor communication mechanism is needed which provides the necessary bandwidth between processors while meeting the requirements discussed above. There are several candidate architectures,including the fully-interconnected system in which each processor is connected to every other processor. Alternative architectures are shared memory,in which all processors interface with a common memory, and a bus oriented structure,in which all processors interface with the bus network, usingeither a single bus or a system of redundant buses. Recently at CDC another form of communication between processors has been studied, called the Ring system. The Ring interconnect system has most often been used in low data rate applications such as telephone networks, minicomputer interconnection systems, and in peripheral I/O systems. CDC has been experimenting in using the Ring to tightly interconnect high performance processors. Several examples of candidate Ring system configurations are shown in Figure 4. With the Ring system architecture the data is passed from one processor to another and circulates around the Ring until being taken off by the intended processor. In the CDC design the data does not pass through the processors but instead passes through a piece of hardware called the Ring Port which makes decisions regarding the passage of the data. The Ring shifts simultaneously so that multiple data packets can exist on the Ring simultaneously. Effectively then the data bandwidth is multiplied by the number of processors on the ring, providing that the algorithm or computational work can be structured appropriately. In the signal processing applications which CDC has investigated, the algorithm can generally be partitioned so that the main data flows take place between adjacent processors on a ring. The Ring system represents a relatively low-cost, very high-bandwidth mechanism for passing data between processors. It tends to have a very long access time, however, when data needs to be passed from one processor to another which is a long way around the ring. Therefore, algorithms in which such passages of data would be required, are probably not appropriate for the Ring system. In system architectures which CDC is exploring, a configuration is envisioned in which capability for rapid interaction between system elements is provided through shared memory in addition to the Ring.

The form of the Ring system which CDC has been investigating most closely is shown in the lower left hand corner of Figure 4. This form has two counter-rotating rings to which all processors are connected, and in which data flows are in opposite directions. This system can provide 1.4 megabits per second of data and control flow in each direction.

CONCLUSION

The cooperative research project between CDC and CMU is developing new problem-oriented, high-speed, digital processor architectures for image processing. The current project is developing a processor capable of 0.2 billion instructions per second. An array of 50 such processors would provide a processing capability of 10 billion operations per second. A complete system would have an hierarchy of memory including high-density, magnetic recorders capable of 100 to 200 megabits per second; and CCD memory, roughly in the range of 10^9 megabits.

We at CDC appreciate the opportunity to work with the staff at CMU. The project has benefited from this joint interaction between industry and university. It needs to be mentioned that this paper represents the collective efforts of a number of engineers, programmers, technicians and management personnel at Control Data. We at CDC look forward to continued interaction with members of the CMU staff which include: Raj Ready, Bob Hon, Steve Rubin, Steve Saunders, and Bob Sproull.

TABLE 1 Functional Unit Characteristics

Unit	No.	Operations	Data Types *
Adder	2	1's Complement 2's Complement Increment Double Merge Bytes	Word Dual Byte Double Word **
Multiplier	1	2's Complement Magnitude Cross-Byte	Word Dual Byte
Shift/Boolean	1	Right Shift Right Circulate 0-15 Positions 16 Boolean FNS.	Word Double Word **
File	1	Simultaneous Read & Write 8 Word Capacity	Dual Word Double Word
Data Memories	2	Read or Write 1024 Word Capacity 16 Indices Direct Indirect Post-Increment Post-Decrement Post-Add Constant Post-Subtract Constant Address Compare	Word Double Word **
Ring Port	1	Ring I/O 16 Word Input Buffer 16 Word Output Buffer Connects to all Files, and Switches	Word

* 16 Bits per word, 8 Bits per Byte

** With two Functional Units of this type

TABLE 2 Bandwidth Measures of SPARC Performance

Mechanism	Bits/Second
Internal Data Transfer	12.8 (10^9)
Internal Microcontrol	10 (10^9)
Ring Port (each)	1.4 (10^9)
Memory Port (each)	0.8 (10^9)

TABLE 3 SPARC Performance Characteristics
(Millions of Operations Per Second)

Operations per sec.	Data Format			
	Fixed-Point			Floating-Point
	8-b	16-b	32-b	64-bits
Add/Subtract	200	100	50	12
Multiplications	100	50	10	12
Shift/Boolean	100	50	25	12
File Manipulations	400	200	100	50
Memory Read/Write	400	200	100	50
Input/Output	400	200	100	50
Comparisons	1200	600	100	-
TOTAL ARITHMETIC OPS	400	200	85	36

TABLE 4 MICROCODE ASSEMBLER

- FREE FORMAT INPUT
- PRODUCE ABSOLUTE/RELOCATABLE BINARY
- CONDITIONAL ASSEMBLY
- PROGRAM STATISTICS
- SYMBOL AND FUNCTIONAL UNIT CROSS REFERENCE
- BATCH OR INTERACTIVE
- LOGICAL DIAGNOSTICS
- DATA FILE INITIALIZATION

TABLE 5 General Source Format

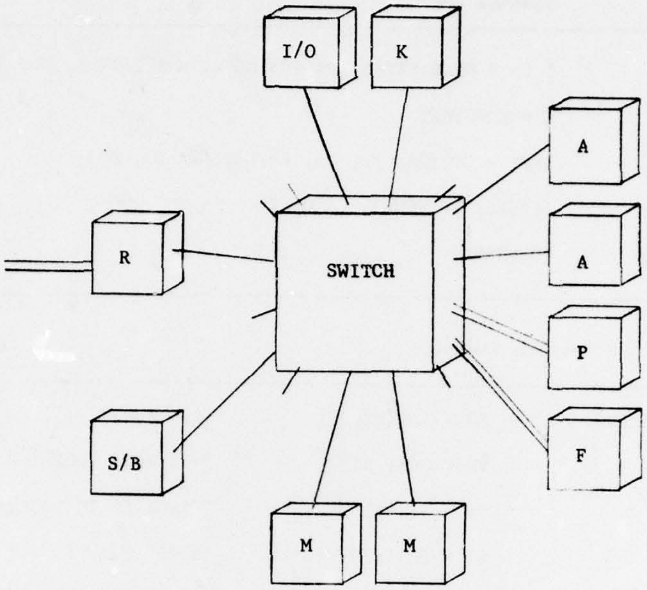
FIELD TYPE	GENERAL FORM
LABEL	1 to 8 CHARACTERS, BEGINNING WITH A LETTER, EG. PART 2
CONSTANT	K = CONSTANT
MAP	DEST = OP (RA, RB, RC, RD/C1, C2, C3, C4)
JUMP	JA(CLK) IF (OPR1, R, OPR2)
COMMENT	"COMMENT"

TABLE 6 Microinstruction Examples

TOP	K = $3B27	A0 = ADD(D2,D3)	"ADD SUMS
		F0 = G4X(,B1)	"WRITE G, LOCATION 4
		B0 = *(,A0)	"CLOCK A0 TO BOOLEAN
		B1 = PASF(F0)	"SHIFT FILE F
		F0 = F5XGX7(B0,B1)	"MOVE TEMP VARIABLES
		A0 = *(L0)	"ADD CROSS PRODUCTS
		A1 = *(H0)	" DITTO
	K = SUB2	JK(PUSH)	"JUMP TO SUBROUTINE SUB2

TABLE 7 Candidate Applications for Dedicated Processor Arrays

GOVERNMENT	CHANGE DETECTION MAPPING MAN/MACHINE INTERFACE AUTOMATED INFORMATION EXTRACTION FUSION
COMMERCIAL	WEATHER NUCLEAR SEISMIC MEDICAL INDUSTRIAL INSPECTION



A	-	ADDER
F	-	FILE
K	-	CONTROL
I/O	-	INPUT OUTPUT
M	-	MEMORY
P	-	MULTIPLIER
R	-	RING PORT
S/B	-	SHIFT/BOOLEAN

Figure 1. SPARC Organization

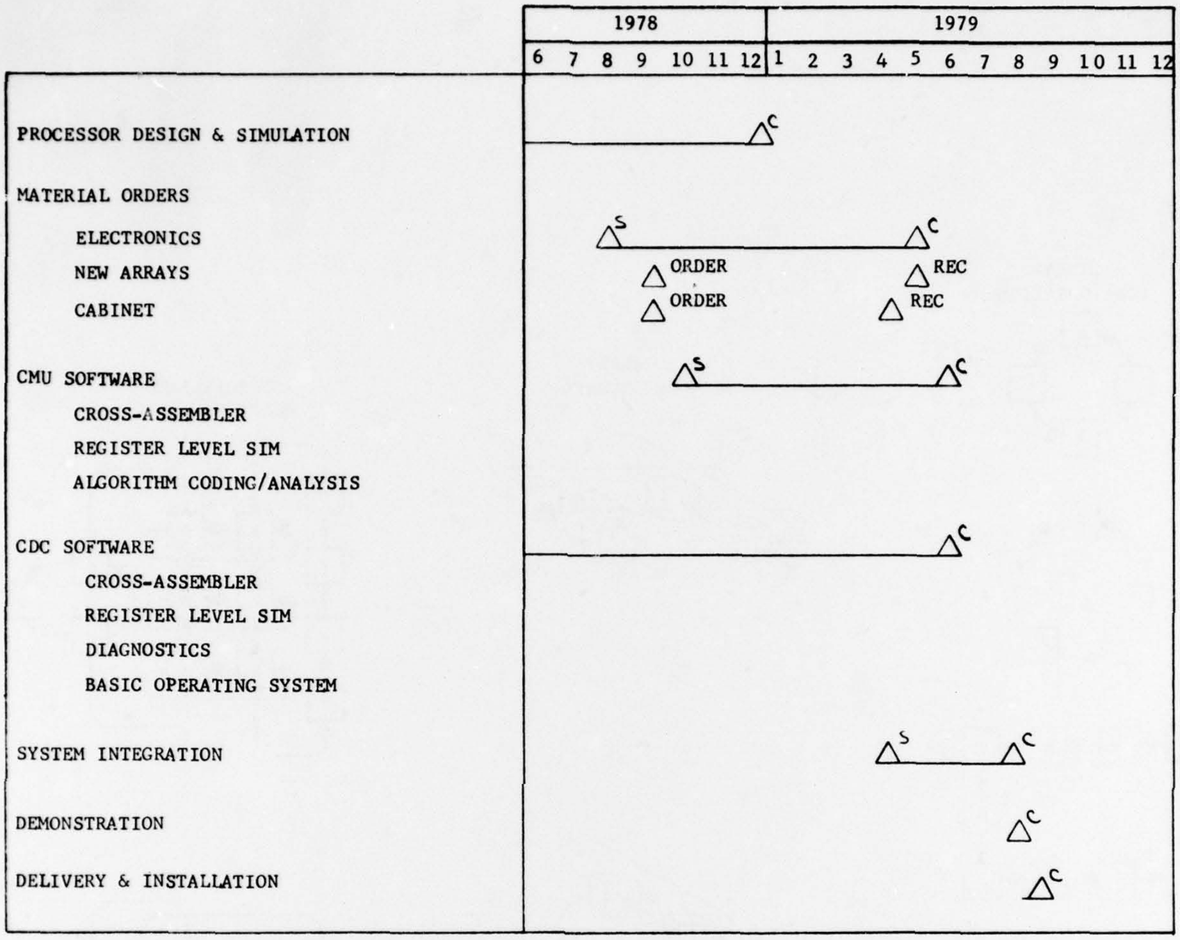


Figure 2. SPARC Development Schedule

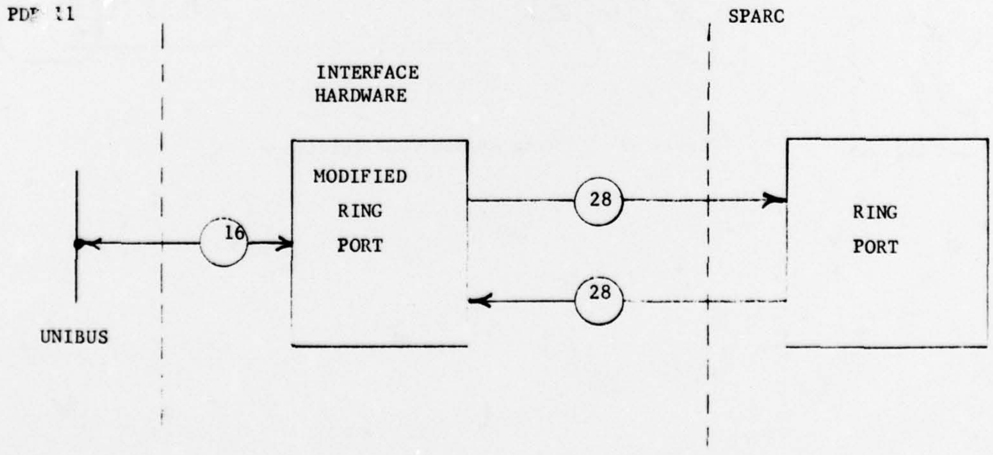


Figure 3. SPARC/PDP-11 INTERFACE

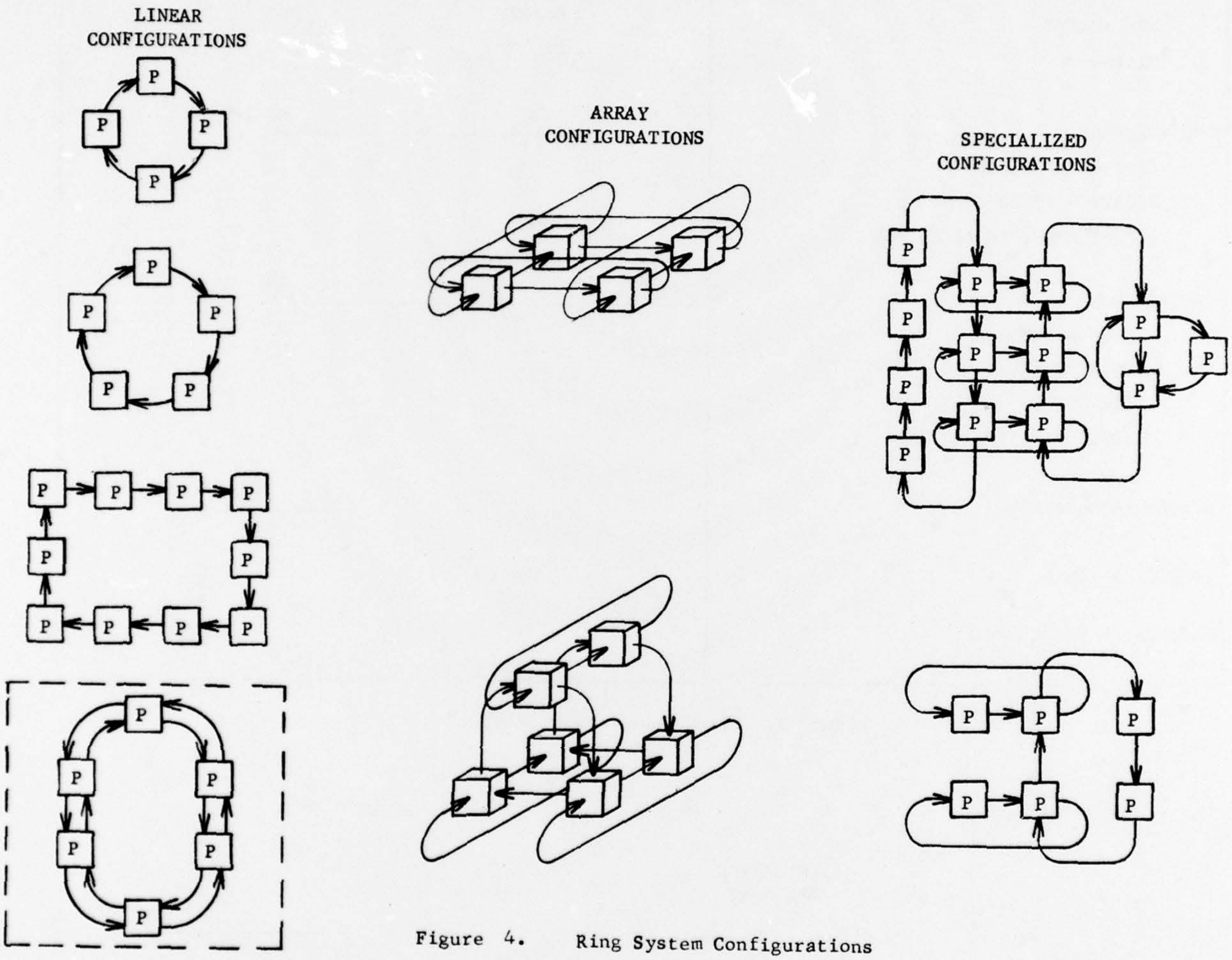


Figure 4. Ring System Configurations

INVESTIGATION OF VLSI TECHNOLOGIES FOR IMAGE PROCESSING

W.L. Eversole, D.J. Mayer, F.B. Frazee, and T.F. Cheek, Jr.

Texas Instruments Incorporated
13500 North Central Expressway, P.O. Box 225936
Dallas, Texas 75265

ABSTRACT

This paper summarizes work to date performed for Carnegie-Mellon University on the investigation of very large scale integration (VLSI) implementations for image processing. Discussion of a real-time image processor concept and the implementation of two complex image processing algorithms are presented.

Introduction

The requirement exists within DoD for an image processor which can provide automatic and semiautomatic interpretation of images. Rapid advances in integrated circuit technology will make possible the realization of highly complex image processing functions on a monolithic substrate.

The thrust of this study effort is to investigate very large scale integration implementations of real-time processing algorithms for potential image processing applications.

Signal processing of the complexity required for a real-time, compact, low-cost image processing has in the past been impractical, if not impossible, because of component technology limitations in the field of processing electronics. However, a new era, one in which computational capability and capacity should outpace the development of algorithmic methods of implementing complex image processors, is near. Tremendous strides made recently in electronic processor development and related componentry present new freedoms in algorithm implementation.

Over the last 20 years, the semiconductor industry has progressed steadily in its efforts to get more capability from solid state technologies at lower costs. Products of this effort include increased functional densities (more capability in smaller volume), improved performance/power ratios, higher processing throughput rates, and improved reliability.

Many digital semiconductor technologies foster new generations of products and product families that, in turn, lead to new realms of applications. No matter how phenomenal this development pace seems, there is no reason to believe that these trends will not continue. For example, the state-of-the-art in active digital element groups per chip has moved from the small-scale integration (SSI) phase of the mid-1960's to today's large-scale integration (LSI) devices. VLSI will be tomorrow's standard technology and is developing rapidly. Recent developments in LSI, VLSI, and other unique component developments allow concentrated algorithm computing power without the former penalties of execution time, size, power and cost. These advancements could result in the implementation of real-time arithmetic logic units (ALUs) of the complexity required for image processing. Furthermore, an understanding of the potential for implementing complex algorithms with minaturized hardware provides the necessary tie between research and digital integrated circuit (IC) development efforts. An example of how these IC developments could be incorporated in an image processor is described.

DIGITAL VLSI IMAGE PROCESSOR

The concept of a VLSI implementation of a digital image processor based on multiple ALUs and buffer memories is shown in Figure 1. The buffer memories accept single line video data and format the data for processing by one or more on-chip ALUs which operate simultaneously on the imagery. Several blocks of buffer memory are included to process images of various resolutions. Each ALU performs a separate image processing function.

Several image processing algorithms are under investigation including algorithms for image enhancement, image restoration, feature extraction, and image bandwidth reduction. Two image processing functions; median filtering for noise suppression and image compression using block truncation coding[1] techniques are discussed below.

Median Filters

Median filtering is a nonlinear signal processing technique used for noise suppression in images. The median filter consists of a sliding window encompassing an odd number of pixels. The center pixel in the window is replaced by the median of the pixels within the window. The median pixel value is that pixel value for which half of the pixel values are smaller or equal in value and half are larger or equal in value. Median filtering is more effective in reducing the effect of discrete impulse noise than smoothly generated noise.[2]

An efficient technique for determining the median has been developed at Carnegie-Mellon University. An example of a one-dimensional median filter using this technique is shown in Figure 2 for three input signals, A, B, and C. First A and B are compared and the larger of A and B is placed on the top line while the smaller of A and B is placed on the middle line. Next the smaller of A and B, and C are compared. Again the larger value is placed on the top line of the two lines being compared. The last comparison is made between the larger of A and B and the larger of C and the smaller of A and B. This median operator requires three comparators and the median values always appear on the middle line. This approach can be applied to five signals as shown in Figure 3. After the first three comparisons the top line can be eliminated from consideration because this line is the larger of A, B, C, and D and cannot be the median value. The fourth comparator eliminates the fourth line since this line is the smaller of A, B, C and D and cannot be the median. Now only three lines are left and the median value is found as in Figure 2. Figure 3 requires seven comparators to implement. This technique can be extended to larger window sizes.

For larger two-dimensional median filters Carnegie-Mellon University has suggested finding an approximation of the median of a 5 x 5 array by finding the median of only five pixels at a time and then using these median values as inputs to a sixth median filter to find the "median of medians". Carnegie-Mellon University has performed statistical analysis on this operation and found that approximately 70 percent of the time the resulting median is either the 12th, 13th, or 14th value of the 5 x 5 array. No analysis has yet been performed to determine if this approximation is accurate enough for image processing.

For the case of five inputs, a second technique was developed which results in a more efficient digital implementation of the median operator as shown in Figure 4. Although this technique requires one more comparator than the technique of Figure 3, parallel processing reduces the total number of gates required and reduces the computation time. However, this approach does not extend to larger filter sizes.

In Figure 4 five 8-bit numbers are loaded into a register file containing five individually addressable 8-bit registers. These five numbers are then tested in pairs by the magnitude comparators to determine the greater binary numerical values of each pair. By using five comparisons it can be shown that in the worst case two of the five numbers can be eliminated from the median location process. The "3 of 5" logic is a combinational circuit that determines from the comparison tests which of the five numbers are to be processed further. The three numbers are then selected from the 5 x 8 register file by the two 3:1 multiplexers and the one 5:1 multiplexer. These three numbers are then stored in the 3 x 8 register file and are processed in a similar manner by three more magnitude comparators. From these tests the median number can be determined and the "1 of 3" logic controls the 3:1 multiplexer to allow the median to pass through the system.

The estimated gate count for this ALU is 1600, with a maximum delay path of 22 gates. For such a system to operate at a 10 MHz video data rate, each gate element can have a delay of no more than 4.5 nsec.

A block diagram of the implementation of the approximation of the median is shown in Figure 5. This is a pipeline approach using a shift register to buffer the output of the first median operator. The second median operator determines the "median of medians".

Block Truncation Coding

Several techniques in image processing require computation of the mean and/or variance of a block of pixels. A recent application is a bandwidth compression scheme developed at Purdue University called Block Truncation Coding.[1] In this technique, the sample mean and variance of small blocks of an image are used to statistically reconstruct the image from binarized image blocks. The following equations define the sample mean and variance, respectively

$$\overline{X} = 1/N \sum_{i=1}^{N} X_i$$

$$\overline{\sigma}^2 = 1/(N-1) \sum_{i=1}^{N} (X_i - \overline{X})^2$$

A digital implementation of the block truncation encoding algorithm for 4 by 4 pixel blocks has been investigated and is shown in Figure 6. The input data is loaded into an accumulator which computes the mean of each block of 16 pixels. A control bit identifies the first word of each block. The data is also input to a shift register which delays the data for the variance and binarization operations until the mean is calculated. The mean is loaded into a delay register for output. A magnitude comparator operates on the delayed input signal and the mean in order to binarize the data, i.e., if a data point is greater than the mean it is binarized to "1" otherwise it is "0". The binary data is input to a shift register which holds the data for output. The variance computation is calculated in parallel with the binarization. The mean is subtracted from the input data and the result is squared and input to an accumulator which completes the variance calculation. An output formatter accepts the 16 binarized data bits, the variance, and the mean and formats the data as desired.

The estimated gate count for this ALU is 3800, with a maximum delay path of 30 gates. For 10 MHz operation, each gate element can have a delay of no more than 3.3 nsec.

CONCLUSIONS

This paper discussed the concept of a digital VLSI image processor containing multiple arithmetic logic units and buffer memory needed to implement several image processing algorithms. The preliminary digital design of a median operator for a 5 x 5 pixel window and 8-bit accuracy was described. Also a digital design capable of computing the mean and standard deviation and performing binarization on a 4 x 4 pixel block with 8-bit accuracy was presented. Investigation of digital integrated circuit implementation of the appropriate buffer memories and other algorithms for realizing a image processor is continuing.

REFERENCES

1. O.R. Mitchell, E.J. Delp, and S.G. Carlton, "Block Truncation Coding: A New Approach to Image Compression", Conference Record, 1978 IEEE International Conference on Communications, pp. 12B.1.1 - 12B.1.4.

2. Pratt, W.K., Digital Image Processing, Wiley-Interscience, New York, 1978.

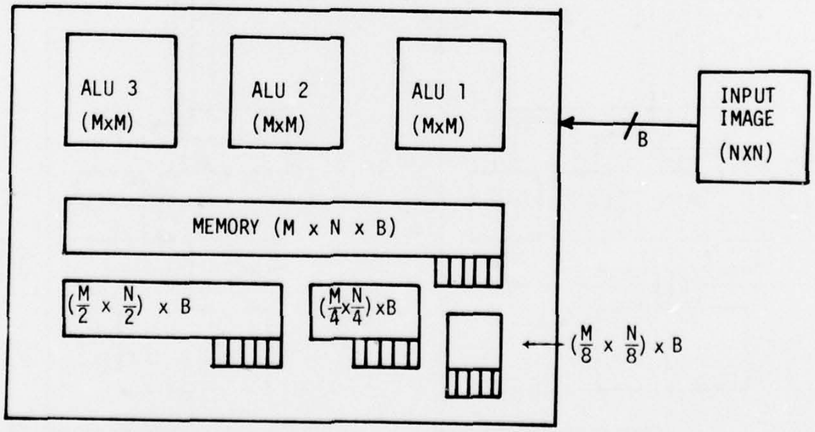

ALU 3
(MxM)
ALU 2
(MxM)
ALU 1
(MxM)
INPUT
IMAGE
(NXN)
B
MEMORY (M x N x B)
$(\frac{M}{2} \times \frac{N}{2}) \times B$
$(\frac{M}{4} \times \frac{N}{4}) \times B$
$(\frac{M}{8} \times \frac{N}{8}) \times B$

Figure 1. Block Diagram of A VLSI Image Processor

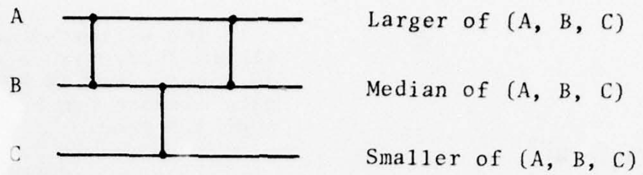


Figure 2. Median Operation for Three Signals

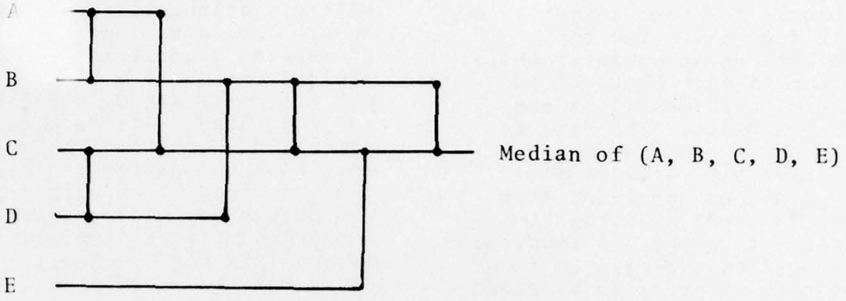


Figure 3. Median Operation for Five Signals

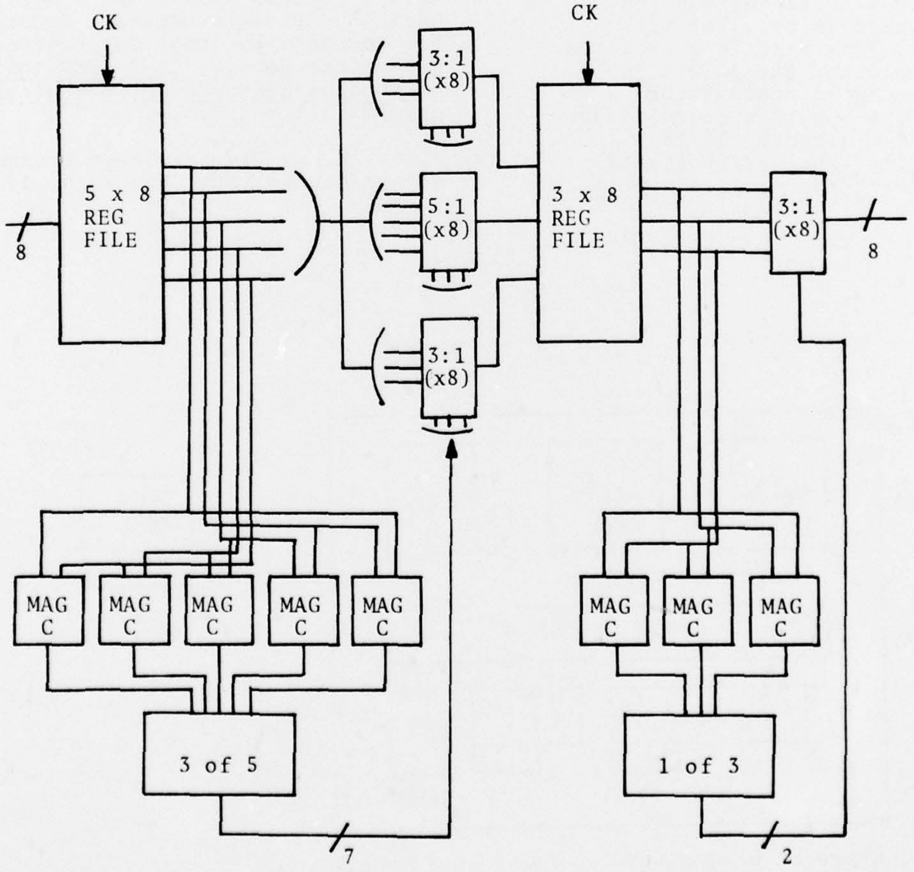


Figure 4. Digital Implementation of Median Operator for Five Input Signals

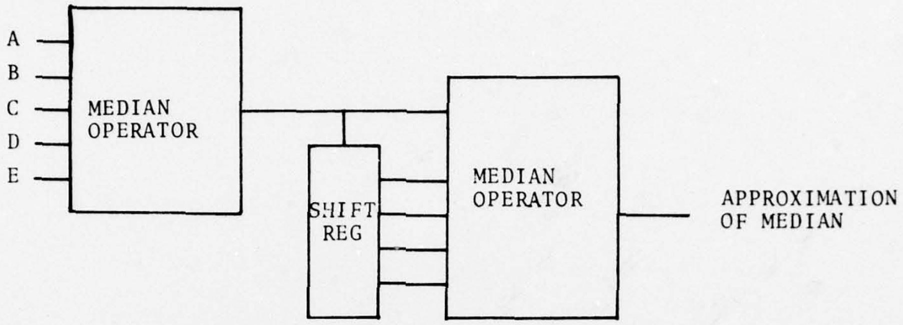


Figure 5. Implementation of Approximation of Median for 5 x 5 Window Size

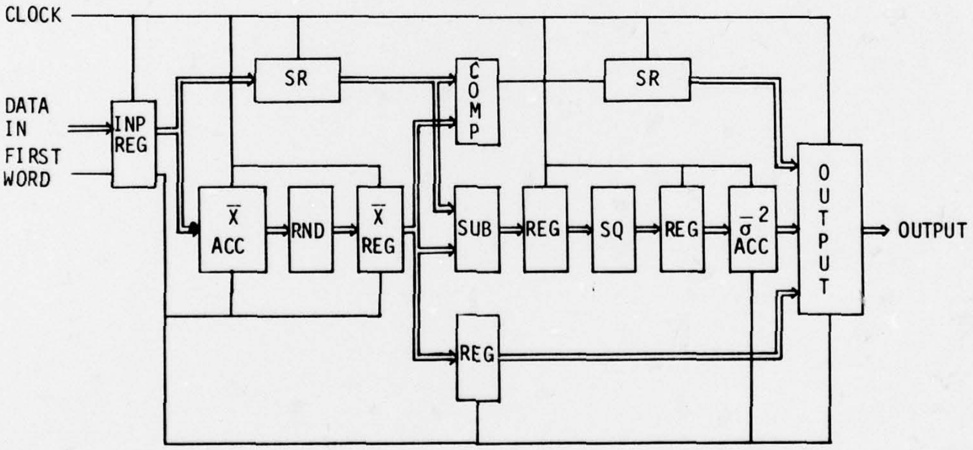


Figure 6. Digital Implementation of Block Truncation Encoding